Cluster Server 7.4.2 Configuration and Upgrade Guide - Linux

Last updated: 2020-08-18

Legal Notice

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

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

Veritas Technologies LLC
2625 Augustine Drive
Santa Clara, CA 95054
http://www.veritas.com

Technical Support

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

You can manage your Veritas account information at the following URL:
https://my.veritas.com

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

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

Documentation

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

Documentation feedback

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

You can also see documentation information or ask a question on the Veritas community site:
http://www.veritas.com/community/

Veritas Services and Operations Readiness Tools (SORT)

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

Contents

Section 1

Configuring Cluster Server using the script-based installer

I/O fencing requirements

This chapter includes the following topics:

- I/O fencing requirements

I/O fencing requirements

Depending on whether you plan to configure disk-based fencing or server-based fencing, make sure that you meet the requirements for coordination points:

- Coordinator disks
  See "Coordinator disk requirements for I/O fencing" on page 15.
- CP servers
  See "CP server requirements" on page 16.

To configure disk-based fencing or to configure server-based fencing with at least one coordinator disk, make sure a version of Veritas Volume Manager (VxVM) that supports SCSI-3 persistent reservations (SCSI-3 PR) is installed on the VCS cluster.

See the *Veritas InfoScale™ Installation Guide*.

If you have installed Veritas InfoScale Enterprise in a virtual environment that is not SCSI-3 PR compliant, review the requirements to configure non-SCSI-3 fencing.

See "Non-SCSI-3 I/O fencing requirements" on page 19.

Coordinator disk requirements for I/O fencing

Make sure that the I/O fencing coordinator disks meet the following requirements:

- For disk-based I/O fencing, you must have at least three coordinator disks or there must be odd number of coordinator disks.
- The coordinator disks must be DMP devices.
- Each of the coordinator disks must use a physically separate disk or LUN.

Veritas recommends using the smallest possible LUNs for coordinator disks.

- Each of the coordinator disks should exist on a different disk array, if possible.
- The coordinator disks must support SCSI-3 persistent reservations.
- Coordinator devices can be attached over iSCSI protocol but they must be DMP devices and must support SCSI-3 persistent reservations.
- Veritas recommends using hardware-based mirroring for coordinator disks.
- Coordinator disks must not be used to store data or must not be included in disk groups that store user data.
- Coordinator disks cannot be the special devices that array vendors use. For example, you cannot use EMC gatekeeper devices as coordinator disks.
- The coordinator disk size must be at least 128 MB.

CP server requirements

VCS 7.4.2 clusters (application clusters) support coordination point servers (CP servers) that are hosted on the following VCS and SFHA versions:

- VCS 6.1 or later single-node cluster
- SFHA 6.1 or later cluster

Upgrade considerations for CP servers

- Upgrade VCS or SFHA on CP servers to version 7.4.2 if the current release version is prior to version 6.1.
- You do not need to upgrade CP servers to version 7.4.2 if the release version is 6.1 or later.
- CP servers on version 6.1 or later support HTTPS-based communication with application clusters on version 6.1 or later.
- CP servers on version 6.1 to 7.0 support IPM-based communication with application clusters on versions before 6.1.
- You need to configure VIPs for HTTPS-based communication if release version of application clusters is 6.1 or later.

Make sure that you meet the basic hardware requirements for the VCS/SFHA cluster to host the CP server.

See the *Veritas InfoScale™ Installation Guide*.

Note: While Veritas recommends at least three coordination points for fencing, a single CP server as coordination point is a supported server-based fencing configuration. Such single CP server fencing configuration requires that the coordination point be a highly available CP server that is hosted on an SFHA cluster.

Make sure you meet the following additional CP server requirements which are covered in this section before you install and configure CP server:

- Hardware requirements
- Operating system requirements
- Networking requirements (and recommendations)
- Security requirements

Table 1-1 lists additional requirements for hosting the CP server.

Table 1-1 CP server hardware requirements

Hardware required	Description
Disk space	To host the CP server on a VCS cluster or SFHA cluster, each host requires the following file system space: ■ 550 MB in the /opt directory (additionally, the language pack requires another 15 MB) ■ 300 MB in /usr ■ 20 MB in /var ■ 10 MB in /etc (for the CP server database)
Storage	When CP server is hosted on an SFHA cluster, there must be shared storage between the nodes of this SFHA cluster.
RAM	Each CP server requires at least 512 MB.
Network	Network hardware capable of providing TCP/IP connection between CP servers and VCS clusters (application clusters).

Table 1-2 displays the CP server supported operating systems and versions. An application cluster can use a CP server that runs any of the following supported operating systems.

Table 1-2 CP server supported operating systems and versions

CP server	Operating system and version
CP server hosted on a VCS single-node cluster or on an SFHA cluster	CP server supports any of the following operating systems: ■ Linux: ■ RHEL 7.7 ■ RHEL 8.1 ■ SLES 12 ■ SLES 15 ■ CentOS 7.7 ■ CentOS 8.1 ■ OL 7.7 Review other details such as supported operating system levels and architecture for the supported operating systems. See the *Veritas InfoScale 7.4.2 Release Notes* for that platform.

Following are the CP server networking requirements and recommendations:

- Veritas recommends that network access from the application clusters to the CP servers should be made highly-available and redundant. The network connections require either a secure LAN or VPN.
- The CP server uses the TCP/IP protocol to connect to and communicate with the application clusters by these network paths. The CP server listens for messages from the application clusters using TCP port 443 if the communication happens over the HTTPS protocol. TCP port 443 is the default port that can be changed while you configure the CP server.

  Veritas recommends that you configure multiple network paths to access a CP server. If a network path fails, CP server does not require a restart and continues to listen on all the other available virtual IP addresses.
- When placing the CP servers within a specific network configuration, you must take into consideration the number of hops from the different application cluster nodes to the CP servers. As a best practice, Veritas recommends that the number of hops and network latency from the different application cluster nodes to the CP servers should be equal. This ensures that if an event occurs that results in an I/O fencing scenario, there is no bias in the race due to difference in number of hops or network latency between the CPS and various nodes.

For communication between the VCS cluster (application cluster) and CP server, review the following support matrix:

For information about establishing secure communications between the application cluster and CP server, see the *Cluster Server Administrator's Guide*.

Non-SCSI-3 I/O fencing requirements

Supported virtual environment for non-SCSI-3 fencing:

- VMware Server ESX 4.0, 5.0, 5.1, and 5.5 on AMD Opteron or Intel Xeon EM64T (x86_64)
  Guest operating system: See the *Veritas InfoScale Release Notes* for the list of supported Linux operating systems.

Make sure that you also meet the following requirements to configure fencing in the virtual environments that do not support SCSI-3 PR:

- VCS must be configured with Cluster attribute UseFence set to SCSI3
- For server-based I/O fencing, all coordination points must be CP servers

Preparing to configure VCS clusters for data integrity

This chapter includes the following topics:

About planning to configure I/O fencing

After you configure VCS with the installer, you must configure I/O fencing in the cluster for data integrity. Application clusters on release version 7.4.2 (HTTPS-based communication) only support CP servers on release version 6.1 and later.

You can configure disk-based I/O fencing, server-based I/O fencing, or majority-based I/O fencing. If your enterprise setup has multiple clusters that use VCS for clustering, Veritas recommends you to configure server-based I/O fencing.

The coordination points in server-based fencing can include only CP servers or a mix of CP servers and coordinator disks.

Veritas also supports server-based fencing with a single coordination point which is a single highly available CP server that is hosted on an SFHA cluster.

Warning: For server-based fencing configurations that use a single coordination point (CP server), the coordination point becomes a single point of failure. In such configurations, the arbitration facility is not available during a failover of the CP server in the SFHA cluster. So, if a network partition occurs on any application cluster during the CP server failover, the application cluster is brought down. Veritas recommends the use of single CP server-based fencing only in test environments.

You use majority fencing mechanism if you do not want to use coordination points to protect your cluster. Veritas recommends that you configure I/O fencing in majority mode if you have a smaller cluster environment and you do not want to invest additional disks or servers for the purposes of configuring fencing.

Note: Majority-based I/O fencing is not as robust as server-based or disk-based I/O fencing in terms of high availability. With majority-based fencing mode, in rare cases, the cluster might become unavailable.

If you have installed VCS in a virtual environment that is not SCSI-3 PR compliant, you can configure non-SCSI-3 fencing.

See Figure 2-2 on page 23.

Figure 2-1 illustrates a high-level flowchart to configure I/O fencing for the VCS cluster.

Figure 2-1 Workflow to configure I/O fencing

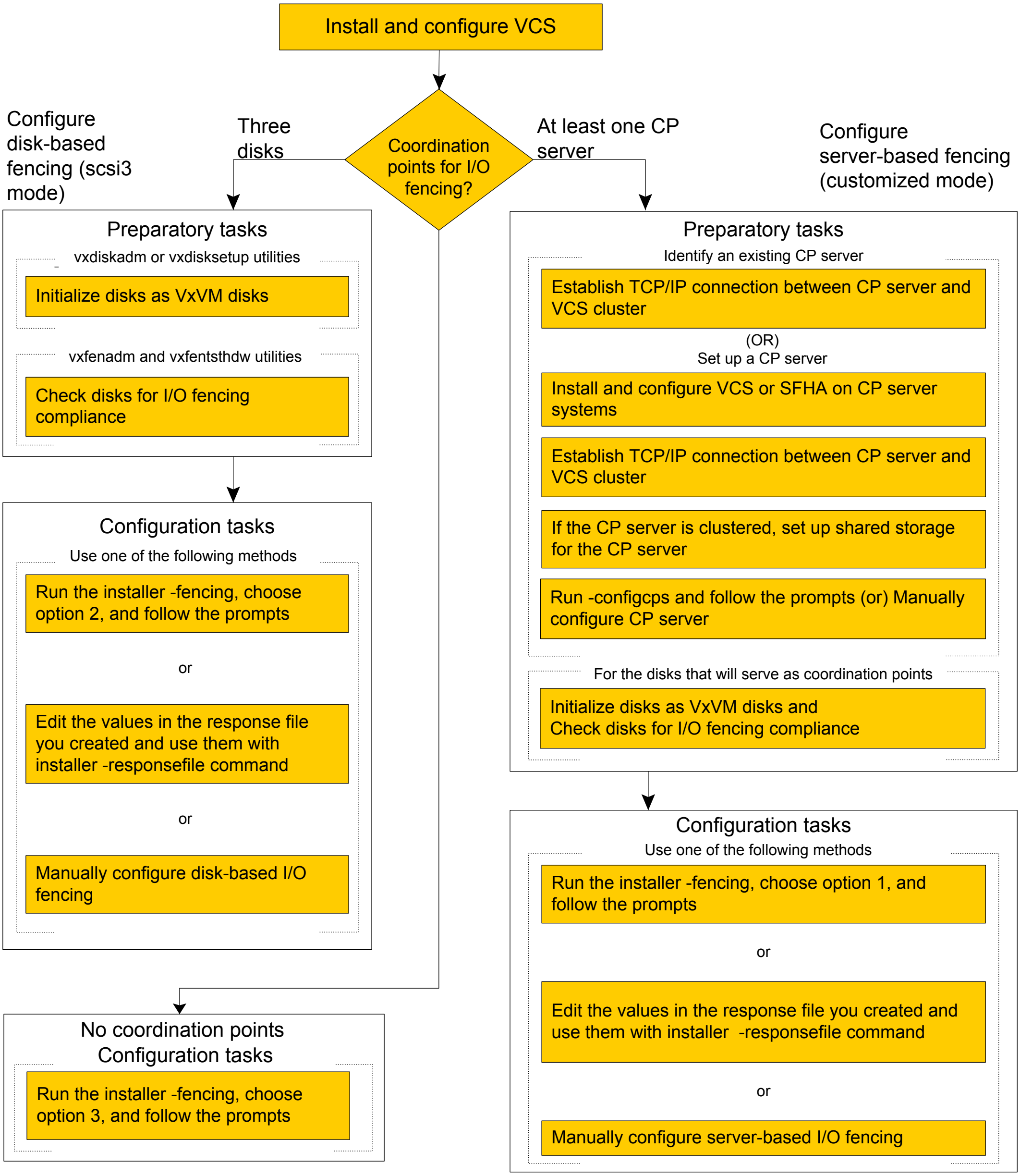
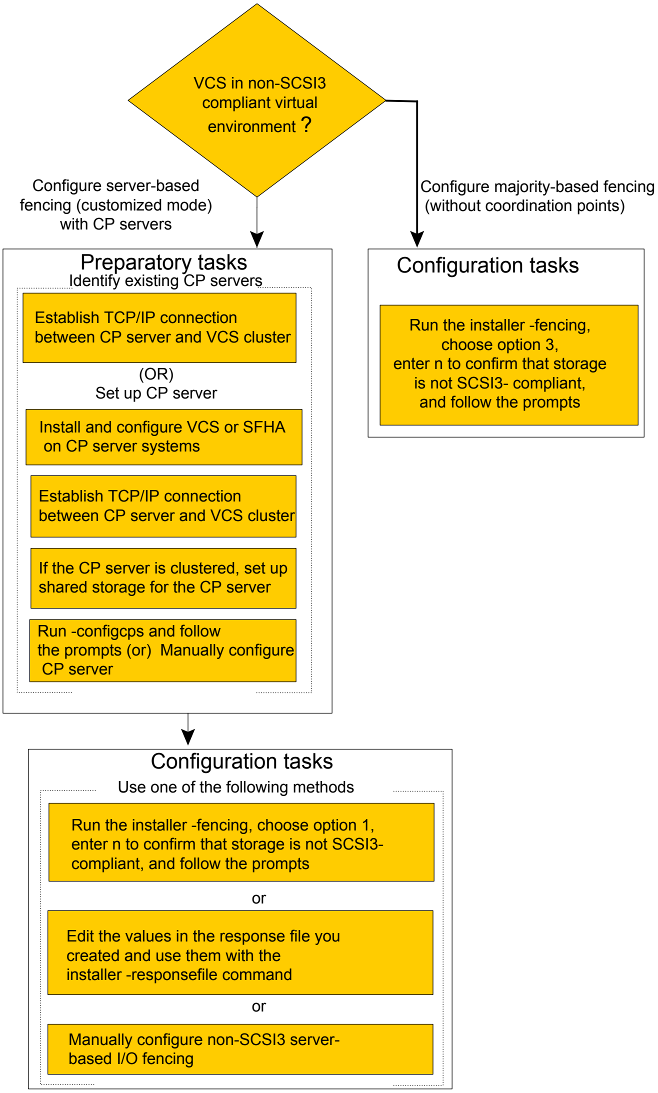

Install and configure VCS
Coordination points for I/O fencing?
Three disks
At least one CP server
Configure disk-based fencing (scsi3 mode)
Configure server-based fencing (customized mode)
Preparatory tasks
vxdiskadm or vxdisksetup utilities
Initialize disks as VxVM disks
vxfenadm and vxfentsthdw utilities
Check disks for I/O fencing compliance
Configuration tasks
Use one of the following methods
Run the installer -fencing, choose option 2, and follow the prompts
or
Edit the values in the response file you created and use them with installer -responsefile command
or
Manually configure disk-based I/O fencing
No coordination points
Configuration tasks
Run the installer -fencing, choose option 3, and follow the prompts
Preparatory tasks
Identify an existing CP server
Establish TCP/IP connection between CP server and VCS cluster
(OR)
Set up a CP server
Install and configure VCS or SFHA on CP server systems
Establish TCP/IP connection between CP server and VCS cluster
If the CP server is clustered, set up shared storage for the CP server
Run -configcps and follow the prompts (or) Manually configure CP server
For the disks that will serve as coordination points
Initialize disks as VxVM disks and
Check disks for I/O fencing compliance
Configuration tasks
Use one of the following methods
Run the installer -fencing, choose option 1, and follow the prompts
or
Edit the values in the response file you created and use them with installer -responsefile command
or
Manually configure server-based I/O fencing

Figure 2-2 illustrates a high-level flowchart to configure non-SCSI-3 I/O fencing for the VCS cluster in virtual environments that do not support SCSI-3 PR.

Figure 2-2 Workflow to configure non-SCSI-3 I/O fencing


VCS in non-SCSI3 compliant virtual environment ?
Configure server-based fencing (customized mode) with CP servers
Configure majority-based fencing (without coordination points)
Preparatory tasks
Identify existing CP servers
Establish TCP/IP connection between CP server and VCS cluster
(OR)
Set up CP server
Install and configure VCS or SFHA on CP server systems
Establish TCP/IP connection between CP server and VCS cluster
If the CP server is clustered, set up shared storage for the CP server
Run -configcps and follow the prompts (or) Manually configure CP server
Configuration tasks
Run the installer -fencing, choose option 3, enter n to confirm that storage is not SCSI3- compliant, and follow the prompts
Configuration tasks
Use one of the following methods
Run the installer -fencing, choose option 1, enter n to confirm that storage is not SCSI3- compliant, and follow the prompts
or
Edit the values in the response file you created and use them with the installer -responsefile command
or
Manually configure non-SCSI3 server- based I/O fencing

After you perform the preparatory tasks, you can use any of the following methods to configure I/O fencing:

Using the installer	See “Setting up disk-based I/O fencing using installer” on page 79. See “Setting up server-based I/O fencing using installer” on page 88. See “Setting up non-SCSI-3 I/O fencing in virtual environments using installer” on page 101. See “Setting up majority-based I/O fencing using installer” on page 103.
Using response files	See “Response file variables to configure disk-based I/O fencing” on page 125. See “Response file variables to configure server-based I/O fencing” on page 129. See “Response file variables to configure non-SCSI-3 I/O fencing” on page 132. See “Response file variables to configure majority-based I/O fencing” on page 134. See “Configuring I/O fencing using response files” on page 124.
Manually editing configuration files	See “Setting up disk-based I/O fencing manually” on page 157. See “Setting up server-based I/O fencing manually” on page 162. See “Setting up non-SCSI-3 fencing in virtual environments manually” on page 176. See “Setting up majority-based I/O fencing manually ” on page 182.

You can also migrate from one I/O fencing configuration to another.

See the *Storage foundation High Availability Administrator's Guide* for more details.

Typical VCS cluster configuration with disk-based I/O fencing

Figure 2-3 displays a typical VCS configuration with two nodes and shared storage. The configuration uses three coordinator disks for I/O fencing.

Figure 2-3 Typical VCS cluster configuration with disk-based I/O fencing

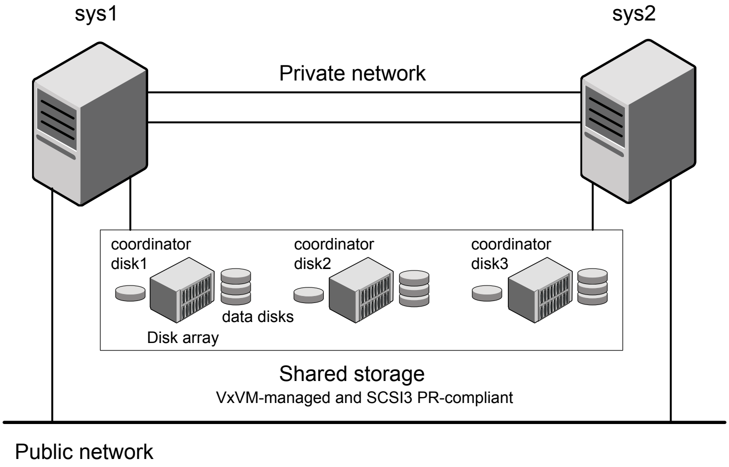
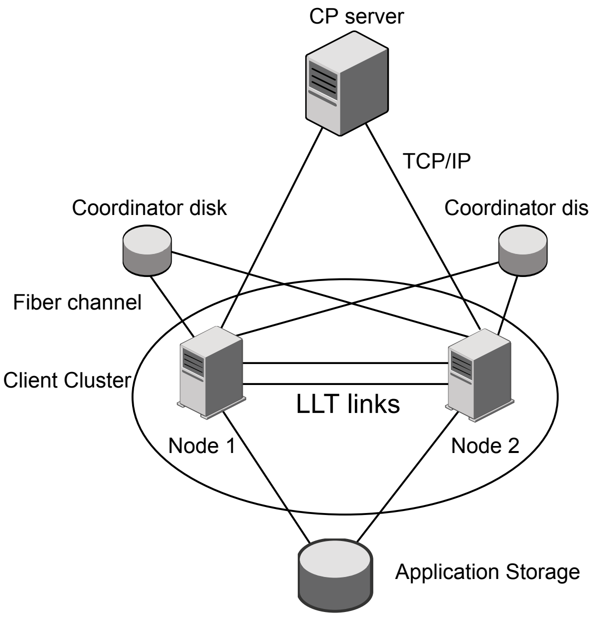


Figure 2-4 displays a configuration using a VCS cluster (with two nodes), a single CP server, and two coordinator disks. The nodes within the VCS cluster are connected to and communicate with each other using LLT links.

Figure 2-4 CP server, VCS cluster, and coordinator disks



Recommended CP server configurations

Following are the recommended CP server configurations:

- Multiple application clusters use three CP servers as their coordination points

  See Figure 2-5 on page 27.

- Multiple application clusters use a single CP server and single or multiple pairs of coordinator disks (two) as their coordination points

  See Figure 2-6 on page 27.

- Multiple application clusters use a single CP server as their coordination point

  This single coordination point fencing configuration must use a highly available CP server that is configured on an SFHA cluster as its coordination point.

  See Figure 2-7 on page 28.

  Warning: In a single CP server fencing configuration, arbitration facility is not available during a failover of the CP server in the SFHA cluster. So, if a network partition occurs on any application cluster during the CP server failover, the application cluster is brought down.

Although the recommended CP server configurations use three coordination points, you can use more than three coordination points for I/O fencing. Ensure that the total number of coordination points you use is an odd number. In a configuration where multiple application clusters share a common set of CP server coordination points, the application cluster as well as the CP server use a Universally Unique Identifier (UUID) to uniquely identify an application cluster.

Figure 2-5 displays a configuration using three CP servers that are connected to multiple application clusters.

Figure 2-5 Three CP servers connecting to multiple application clusters

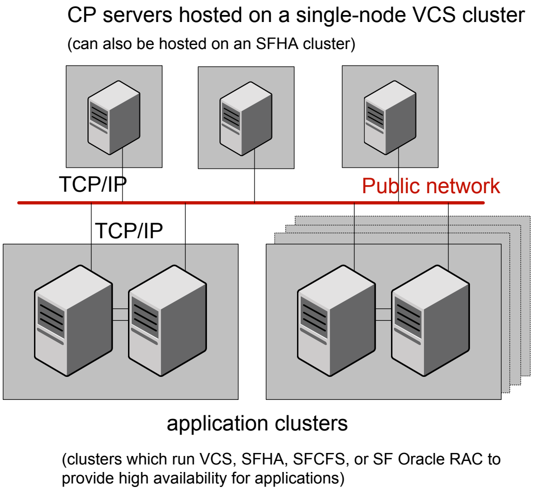

CP servers hosted on a single-node VCS cluster
(can also be hosted on an SFHA cluster)
TCP/IP
Public network
TCP/IP
application clusters
(clusters which run VCS, SFHA, SFCFS, or SF Oracle RAC to provide high availability for applications)

Figure 2-6 displays a configuration using a single CP server that is connected to multiple application clusters with each application cluster also using two coordinator disks.

Figure 2-6 Single CP server with two coordinator disks for each application cluster

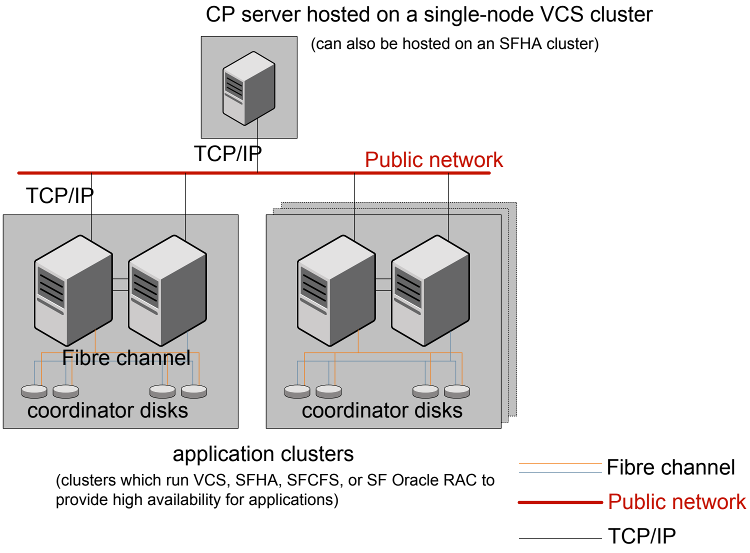
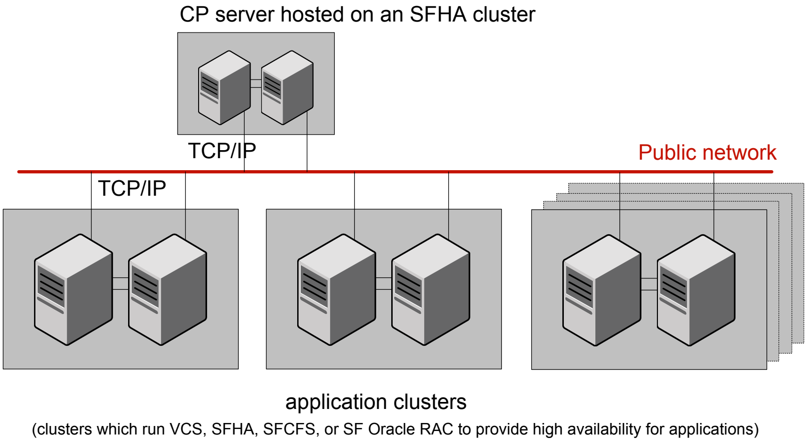

CP server hosted on a single-node VCS cluster
(can also be hosted on an SFHA cluster)
TCP/IP
Public network
TCP/IP
Fibre channel
coordinator disks
coordinator disks
application clusters
(clusters which run VCS, SFHA, SFCFS, or SF Oracle RAC to provide high availability for applications)
Fibre channel
Public network
TCP/IP

Figure 2-7 displays a configuration using a single CP server that is connected to multiple application clusters.

Figure 2-7 Single CP server connecting to multiple application clusters


CP server hosted on an SFHA cluster
TCP/IP
Public network
TCP/IP
application clusters
(clusters which run VCS, SFHA, SFCFS, or SF Oracle RAC to provide high availability for applications)

See "Configuration diagrams for setting up server-based I/O fencing" on page 368.

Setting up the CP server

Table 2-1 lists the tasks to set up the CP server for server-based I/O fencing.

Table 2-1 Tasks to set up CP server for server-based I/O fencing

Task	Reference
Plan your CP server setup	See "Planning your CP server setup" on page 29.
Install the CP server	See "Installing the CP server using the installer" on page 30.
Set up shared storage for the CP server database	See "Setting up shared storage for the CP server database" on page 30.

Table 2-1 Tasks to set up CP server for server-based I/O fencing *(continued)*

Task	Reference
Configure the CP server	See " Configuring the CP server using the installer program" on page 31. See "Configuring the CP server manually" on page 40. See "Configuring CP server using response files" on page 45.
Verify the CP server configuration	See "Verifying the CP server configuration" on page 49.

Planning your CP server setup

Follow the planning instructions to set up CP server for server-based I/O fencing.

To plan your CP server setup

1. Decide whether you want to host the CP server on a single-node VCS cluster, or on an SFHA cluster.

   Veritas recommends hosting the CP server on an SFHA cluster to make the CP server highly available.

2. If you host the CP server on an SFHA cluster, review the following information. Make sure you make the decisions and meet these prerequisites when you set up the CP server:

   - You must set up shared storage for the CP server database during your CP server setup.
   - Decide whether you want to configure server-based fencing for the VCS cluster (application cluster) with a single CP server as coordination point or with at least three coordination points.
     Veritas recommends using at least three coordination points.

3. Set up the hardware and network for your CP server.

   See "CP server requirements" on page 16.

4. Have the following information handy for CP server configuration:

   - Name for the CP server
     The CP server name should not contain any special characters. CP server name can include alphanumeric characters, underscore, and hyphen.
   - Port number for the CP server

Allocate a TCP/IP port for use by the CP server.

Valid port range is between 49152 and 65535. The default port number for HTTPS-based communication is 443.

- Virtual IP address, network interface, netmask, and networkhosts for the CP server

  You can configure multiple virtual IP addresses for the CP server.

Installing the CP server using the installer

Perform the following procedure to install Veritas InfoScale Enterprise and configure VCS or SFHA on CP server systems.

To install Veritas InfoScale Enterprise and configure VCS or SFHA on the CP server systems

◆ Depending on whether your CP server uses a single system or multiple systems, perform the following tasks:

CP server setup uses a single system	Install Veritas InfoScale Enterprise or Veritas InfoScale Availability and configure VCS to create a single-node VCS cluster. Proceed to configure the CP server. See “ Configuring the CP server using the installer program” on page 31. See “Configuring the CP server manually” on page 40.
CP server setup uses multiple systems	Install Veritas InfoScale Enterprise and configure SFHA to create an SFHA cluster. This makes the CP server highly available. See the *Veritas InfoScale Installation Guide* for instructions on installing SFHA. See the *Storage Foundation and High Availability Configuration and Upgrade Guide* for configuring SFHA. Proceed to set up shared storage for the CP server database.

Setting up shared storage for the CP server database

If you configured SFHA on the CP server cluster, perform the following procedure to set up shared storage for the CP server database.

The installer can set up shared storage for the CP server database when you configure CP server for the SFHA cluster.

Veritas recommends that you create a mirrored volume for the CP server database and that you use the VxFS file system type.

To set up shared storage for the CP server database

1 Create a disk group containing the disks. You require two disks to create a mirrored volume.

For example:

```
# vxdg init cps_dg  disk1 disk2
```

2 Create a mirrored volume over the disk group.

For example:

```
# vxassist -g cps_dg make cps_vol volume_size layout=mirror
```

3 Create a file system over the volume.

The CP server configuration utility only supports vxfs file system type. If you use an alternate file system, then you must configure CP server manually.

Depending on the operating system that your CP server runs, enter the following command:

Linux	`# mkfs -t vxfs /dev/vx/rdsk/cps_dg/cps_volume`

Configuring the CP server using the installer program

Use the configcps option available in the installer program to configure the CP server.

Perform one of the following procedures:

For CP servers on single-node VCS cluster:	See "To configure the CP server on a single-node VCS cluster" on page 31.
For CP servers on an SFHA cluster:	See "To configure the CP server on an SFHA cluster" on page 35.

To configure the CP server on a single-node VCS cluster

1 Verify that the `VRTScps` RPM is installed on the node.

2 Run the installer program with the configcps option.

```
# /opt/VRTS/install/installer  -configcps
```

3 Installer checks the cluster information and prompts if you want to configure CP Server on the cluster.

Enter **y** to confirm.

4 Select an option based on how you want to configure Coordination Point server.

```
1) Configure Coordination Point Server on single node VCS system
2) Configure Coordination Point Server on SFHA cluster
3) Unconfigure Coordination Point Server
```

5 Enter the option: [1-3,q] **1**.

The installer then runs the following preconfiguration checks:

- Checks to see if a single-node VCS cluster is running with the supported platform.
  The CP server requires VCS to be installed and configured before its configuration.

The installer automatically installs a license that is identified as a CP server-specific license. It is installed even if a VCS license exists on the node. CP server-specific key ensures that you do not need to use a VCS license on the single-node. It also ensures that Veritas Operations Manager (VOM) identifies the license on a single-node coordination point server as a CP server-specific license and not as a VCS license.

6 Restart the VCS engine if the single-node only has a CP server-specific license.

```
A single node coordination point server will be configured and
VCS will be started in one node mode, do you want to
continue?  [y,n,q]  (y)
```

7 Communication between the CP server and application clusters is secured by using the HTTPS protocol from release 6.1.0 onwards.

Enter the name of the CP Server.

```
Enter the name of the CP Server: [b]   cps1
```

8 Enter valid virtual IP addresses for the CP Server with HTTPS-based secure communication. A CP Server can be configured with more than one virtual IP address.

```
Enter Virtual IP(s) for the CP server for HTTPS,
separated by a space:  [b]  10.200.58.231 10.200.58.232
10.200.58.233
```

Note: Ensure that the virtual IP address of the CP server and the IP address of the NIC interface on the CP server belongs to the same subnet of the IP network. This is required for communication to happen between client nodes and CP server.

---

9 Enter the corresponding CP server port number for each virtual IP address or press **Enter** to accept the default value (443).

```
Enter the default port '443' to be used for all the
virtual IP addresses for HTTPS communication or assign the
corresponding port number in the range [49152, 65535] for
each virtual IP address. Ensure that each port number is
separated by a single
space: [b]  (443) 54442 54443 54447
```

10 Enter the absolute path of the CP server database or press **Enter** to accept the default value (/etc/VRTScps/db).

```
Enter absolute path of the database: [b] (/etc/VRTScps/db)
```

11 Verify and confirm the CP server configuration information.

```
CP Server configuration verification:
-------------------------------------------------
CP Server Name:  cps1
CP Server Virtual IP(s) for HTTPS: 10.200.58.231, 10.200.58.232,
10.200.58.233
CP Server Port(s) for HTTPS: 54442, 54443, 54447
CP Server Database Dir: /etc/VRTScps/db

-------------------------------------------------

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

12 The installer proceeds with the configuration process, and creates a vxcps.conf configuration file.

```
Successfully generated the /etc/vxcps.conf configuration file
Successfully created directory /etc/VRTScps/db on node
```

13 Configure the CP Server Service Group (CPSSG) for this cluster.

```
Enter how many NIC resources you want to configure (1 to 2): 2
```

Answer the following questions for each NIC resource that you want to configure.

14 Enter a valid network interface for the virtual IP address for the CP server process.

```
Enter a valid network interface on sys1 for NIC resource - 1: eth0
Enter a valid network interface on sys1 for NIC resource - 2:  eth1
```

15 Enter the NIC resource you want to associate with the virtual IP addresses.

```
Enter the NIC resource you want to associate with the virtual IP 10.200.58.231 (1 to 2): 1
Enter the NIC resource you want to associate with the virtual IP 10.200.58.232 (1 to 2): 2
```

16 Enter the networkhosts information for each NIC resource.

```
Veritas recommends configuring NetworkHosts attribute to ensure NIC resource
to be always online

Do you want to add NetworkHosts attribute for the NIC device eth0
on system sys1? [y,n,q] y
Enter a valid IP address to configure NetworkHosts for NIC eth0
on system sys1: 10.200.56.22

Do you want to add another Network Host? [y,n,q] n
```

17 Enter the netmask for virtual IP addresses. If you entered an IPv6 address, enter the prefix details at the prompt.

```
Enter the netmask for virtual IP for
HTTPS 192.169.0.220: (255.255.252.0)
```

18. Installer displays the status of the Coordination Point Server configuration. After the configuration process has completed, a success message appears.

    ```
    For example:
    Updating main.cf with CPSSG service group.. Done
    Successfully added the CPSSG service group to VCS configuration.
    Trying to bring CPSSG service group
    ONLINE and will wait for upto 120 seconds

    The Veritas coordination point server is ONLINE

    The Veritas coordination point server has
    been configured on your system.
    ```

19. Run the `hagrp -state` command to ensure that the CPSSG service group has been added.

    ```
    For example:
    # hagrp -state CPSSG
    #Group Attribute System Value
    CPSSG State.... |ONLINE|
    ```

    It also generates the configuration file for CP server (/etc/vxcps.conf). The vxcpserv process and other resources are added to the VCS configuration in the CP server service group (CPSSG).

For information about the CPSSG, refer to the *Cluster Server Administrator's Guide*.

To configure the CP server on an SFHA cluster

1. Verify that the `VRTScps` RPM is installed on each node.
2. Ensure that you have configured passwordless ssh or rsh on the CP server cluster nodes.
3. Run the installer program with the configcps option.

    ```
    # ./installer -configcps
    ```

4. Specify the systems on which you need to configure the CP server.
5. Installer checks the cluster information and prompts if you want to configure CP Server on the cluster.

    Enter **y** to confirm.

6 Select an option based on how you want to configure Coordination Point server.

```
1)  Configure Coordination Point Server on single node VCS system
2)  Configure Coordination Point Server on SFHA cluster
3)  Unconfigure Coordination Point Server
```

7 Enter **2** at the prompt to configure CP server on an SFHA cluster.

The installer then runs the following preconfiguration checks:

- Checks to see if an SFHA cluster is running with the supported platform.
  The CP server requires SFHA to be installed and configured before its configuration.

8 Communication between the CP server and application clusters is secured by HTTPS from Release 6.1.0 onwards.

Enter the name of the CP server.

```
Enter the name of the CP Server: [b]  cps1
```

9 Enter valid virtual IP addresses for the CP Server. A CP Server can be configured with more than one virtual IP address.

```
Enter Virtual IP(s) for the CP server for HTTPS,
 separated by a space: [b] 10.200.58.231 10.200.58.232 10.200.58.233
```

10 Enter the corresponding CP server port number for each virtual IP address or press Enter to accept the default value (443).

```
Enter the default port '443' to be used for all the virtual IP addresses
for HTTPS communication or assign the corresponding port number in the range [49152,
65535] for each virtual IP address. Ensure that each port number is separated by
a single space: [b] (443) 65535 65534 65537
```

11 Enter absolute path of the database.

```
CP Server uses an internal database to store the client information.
As the CP Server is being configured on SFHA cluster, the database should reside
on shared storage with vxfs file system. Please refer to documentation for
information on setting up of shared storage for CP server database.
Enter absolute path of the database: [b] /cpsdb
```

12 Verify and confirm the CP server configuration information.

```
CP Server configuration verification:

CP Server Name: cps1
CP Server Virtual IP(s) for HTTPS: 10.200.58.231, 10.200.58.232,
10.200.58.233
CP Server Port(s) for HTTPS: 65535, 65534, 65537
CP Server Database Dir: /cpsdb

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

13 The installer proceeds with the configuration process, and creates a vxcps.conf configuration file.

```
Successfully generated the /etc/vxcps.conf configuration file
Copying configuration file /etc/vxcps.conf to sys0....Done
Creating mount point /cps_mount_data on sys0. ... Done
Copying configuration file /etc/vxcps.conf to sys0. ... Done
Press Enter to continue.
```

14 Configure CP Server Service Group (CPSSG) for this cluster.

```
Enter how many NIC resources you want to configure (1 to 2): 2

Answer the following questions for each NIC resource that you want to configure.
```

15 Enter a valid network interface for the virtual IP address for the CP server process.

```
Enter a valid network interface on sys1 for NIC resource - 1: eth0
Enter a valid network interface on sys1 for NIC resource - 2: eth1
```

16 Enter the NIC resource you want to associate with the virtual IP addresses.

```
Enter the NIC resource you want to associate with the virtual IP 10.200.58.231 (1 to 2): 1
Enter the NIC resource you want to associate with the virtual IP 10.200.58.232 (1 to 2): 2
```

17 Enter the networkhosts information for each NIC resource.

```
Veritas recommends configuring NetworkHosts attribute to ensure NIC resource
to be always online


Do you want to add NetworkHosts attribute for the NIC device eth0
on system sys1? [y,n,q] y
Enter a valid IP address to configure NetworkHosts for NIC eth0
on system sys1: 10.200.56.22


Do you want to add another Network Host? [y,n,q] n
Do you want to apply the same NetworkHosts for all systems? [y,n,q] (y)
```

18 Enter the netmask for virtual IP addresses. If you entered an IPv6 address, enter the prefix details at the prompt.

```
Enter the netmask for virtual IP for
HTTPS 192.168.0.111: (255.255.252.0)
```

19 Configure a disk group for CP server database. You can choose an existing disk group or create a new disk group.

```
Veritas recommends to use the disk group that has at least
two disks on which mirrored volume can be created.
Select one of the options below for CP Server database disk group:

1)  Create a new disk group
2)  Using an existing disk group

Enter the choice for a disk group: [1-2,q]  2
```

20 Select one disk group as the CP Server database disk group.

```
Select one disk group as CP Server database disk group: [1-3,q] 3
1)  mycpsdg
2)  cpsdg1
3)  newcpsdg
```

21 Select the CP Server database volume.

You can choose to use an existing volume or create new volume for CP Server database. If you chose newly created disk group, you can only choose to create new volume for CP Server database.

```
Select one of the options below for CP Server database volume:
 1)  Create a new volume on disk group newcpsdg
 2)  Using an existing volume on disk group newcpsdg
```

22 Enter the choice for a volume: [1-2,q] **2**.

23 Select one volume as CP Server database volume [1-1,q] **1**

```
1) newcpsvol
```

24 After the VCS configuration files are updated, a success message appears.

```
For example:
Updating main.cf with CPSSG service group .... Done
Successfully added the CPSSG service group to VCS configuration.
```

25 If the cluster is secure, installer creates the softlink `/var/VRTSvcs/vcsauth/data/CPSERVER` to `/cpsdb/CPSERVER` and check if credentials are already present at `/cpsdb/CPSERVER`. If not, installer creates credentials in the directory, otherwise, installer asks if you want to reuse exsting credentials.

```
Do you want to reuse these credentials? [y,n,q] (y)
```

26 After the configuration process has completed, a success message appears.

```
For example:
Trying to bring CPSSG service group ONLINE and will wait for upto 120 seconds
The Veritas Coordination Point Server is ONLINE
The Veritas Coordination Point Server has been configured on your system.
```

27 Run the `hagrp -state` command to ensure that the CPSSG service group has been added.

```
For example:
# hagrp -state CPSSG
#Group Attribute System Value
CPSSG State cps1 |ONLINE|
CPSSG State cps2 |OFFLINE|
```

It also generates the configuration file for CP server (`/etc/vxcps.conf`). The vxcpserv process and other resources are added to the VCS configuration in the CP server service group (CPSSG).

For information about the CPSSG, refer to the *Cluster Server Administrator's Guide*.

Configuring the CP server manually

Perform the following steps to manually configure the CP server.

You need to manually generate certificates for the CP server and its client nodes to configure the CP server for HTTPS-based communication.

Table 2-2 Tasks to configure the CP server manually

Task	Reference
Configure CP server manually for HTTPS-communication	See "Configuring the CP server manually for HTTPS-based communication" on page 41. See "Generating the key and certificates manually for the CP server" on page 42. See "Completing the CP server configuration" on page 45.

Note: If a CP server should support pure IPv6 communication, use only IPv6 addresses in the `/etc/vxcps.conf` file. If the CP server should support both IPv6 and IPv4 communications, use both IPv6 and IPv4 addresses in the configuration file.

Configuring the CP server manually for HTTPS-based communication

Perform the following steps to manually configure the CP server in HTTPS-based mode.

To manually configure the CP server

1. Stop VCS on each node in the CP server cluster using the following command:

   ```
   # hastop -local
   ```

2. Edit the `main.cf` file to add the CPSSG service group on any node. Use the CPSSG service group in the sample main.cf as an example:

   See "Sample configuration files for CP server" on page 280.

   Customize the resources under the CPSSG service group as per your configuration.

3. Verify the `main.cf` file using the following command:

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

   If successfully verified, copy this main.cf to all other cluster nodes.

4. Create the `/etc/vxcps.conf` file using the sample configuration file provided at `/etc/vxcps/vxcps.conf.sample`.

   Veritas recommends enabling security for communication between CP server and the application clusters.

   If you configured the CP server in HTTPS mode, do the following:

   - Edit the `/etc/vxcps.conf` file to set `vip_https` with the virtual IP addresses required for HTTPS communication.
   - Edit the `/etc/vxcps.conf` file to set `port_https` with the ports used for HTTPS communication.

5. Manually generate keys and certificates for the CP server.

   See "Generating the key and certificates manually for the CP server" on page 42.

Generating the key and certificates manually for the CP server

CP server uses the HTTPS protocol to establish secure communication with client nodes. HTTPS is a secure means of communication, which happens over a secure communication channel that is established using the SSL/TLS protocol.

HTTPS uses x509 standard certificates and the constructs from a Public Key Infrastructure (PKI) to establish secure communication between the CP server and client. Similar to a PKI, the CP server, and its clients have their own set of certificates signed by a Certification Authority (CA). The server and its clients trust the certificate.

Every CP server acts as a certification authority for itself and for all its client nodes. The CP server has its own CA key and CA certificate and a server certificate generated, which is generated from a server private key. The server certificate is issued to the Universally Unique Identifier (UUID) of the CP server. All the IP addresses or domain names that the CP server listens on are mentioned in the Subject Alternative Name section of the CP server's server certificate

The OpenSSL library must be installed on the CP server to create the keys or certificates.. If OpenSSL is not installed, then you cannot create keys or certificates. The vxcps.conf file points to the configuration file that determines which keys or certificates are used by the CP server when SSL is initialized. The configuration value is stored in the `ssl_conf_file` and the default value is `/etc/vxcps_ssl.properties`.

To manually generate keys and certificates for the CP server:

1 Create directories for the security files on the CP server.

```
# mkdir -p /var/VRTScps/security/keys /var/VRTScps/security/certs
```

2 Generate an OpenSSL config file, which includes the VIPs.

The CP server listens to requests from client nodes on these VIPs. The server certificate includes VIPs, FQDNs, and host name of the CP server. Clients can reach the CP server by using any of these values. However, Veritas recommends that client nodes use the IP address to communicate to the CP server.

The sample configuration uses the following values:

- Config file name: *https_ssl_cert.conf*
- VIP: *192.168.1.201*
- FQDN: *cpsone.company.com*
- Host name: *cpsone*

Note the IP address, VIP, and FQDN values used in the [alt_names] section of the configuration file are sample values. Replace the sample values with your configuration values. Do not change the rest of the values in the configuration file.

```
[req]
distinguished_name = req_distinguished_name
req_extensions = v3_req

[req_distinguished_name]
countryName = Country Name (2 letter code)
countryName_default = US
localityName = Locality Name (eg, city)
organizationalUnitName = Organizational Unit Name (eg, section)
commonName = Common Name (eg, YOUR name)
commonName_max = 64
emailAddress = Email Address
emailAddress_max = 40

[v3_req]
keyUsage = keyEncipherment, dataEncipherment
extendedKeyUsage = serverAuth
subjectAltName = @alt_names

[alt_names]
DNS.1 = cpsone.company.com
DNS.2 = cpsone
DNS.3 = 192.168.1.201
```

3 Generate a 4096-bit CA key that is used to create the CA certificate.

The key must be stored at `/var/VRTScps/security/keys/ca.key`. Ensure that only root users can access the CA key, as the key can be misused to create fake certificates and compromise security.

```
# /opt/VRTSperl/non-perl-libs/bin/openssl genrsa -out
/var/VRTScps/security/keys/ca.key 4096
```

4 Generate a self-signed CA certificate.

```
# /opt/VRTSperl/non-perl-libs/bin/openssl req -new -x509 -days
days -sha256 -key /var/VRTScps/security/keys/ca.key -subj \

'/C=countryname/L=localityname/OU=COMPANY/CN=CACERT' -out \

/var/VRTScps/security/certs/ca.crt
```

Where, *days* is the days you want the certificate to remain valid, *countryname* is the name of the country, localityname is the city, *CACERT* is the certificate name.

5 Generate a 2048-bit private key for CP server.

The key must be stored at `/var/VRTScps/security/keys/server_private key`.

```
# /opt/VRTSperl/non-perl-libs/bin/openssl genrsa -out \

/var/VRTScps/security/keys/server_private.key 2048
```

6 Generate a Certificate Signing Request (CSR) for the server certificate.

The Certified Name (CN) in the certificate is the UUID of the CP server.

```
# /opt/VRTSperl/non-perl-libs/bin/openssl req -new -sha256 -key
/var/VRTScps/security/keys/server_private.key \

-config https_ssl_cert.conf -subj \

'/C=CountryName/L=LocalityName/OU=COMPANY/CN=UUID' \

-out /var/VRTScps/security/certs/server.csr
```

Where, *countryname* is the name of the country, *localityname* is the city, *UUID* is the certificate name.

7 Generate the server certificate by using the key certificate of the CA.

```
# /opt/VRTSperl/non-perl-libs/bin/openssl x509 -req -days days
-sha256 -in /var/VRTScps/security/certs/server.csr \

-CA /var/VRTScps/security/certs/ca.crt -CAkey \

/var/VRTScps/security/keys/ca.key \

-set_serial 01 -extensions v3_req -extfile https_ssl_cert.conf \

-out /var/VRTScps/security/certs/server.crt
```

Where, *days* is the days you want the certificate to remain valid, *https_ssl_cert.conf* is the configuration file name.

You successfully created the key and certificate required for the CP server.

8 Ensure that no other user except the root user can read the keys and certificates.

9 Complete the CP server configuration.

See "Completing the CP server configuration" on page 45.

Completing the CP server configuration

To verify the service groups and start VCS perform the following steps:

1 Start VCS on all the cluster nodes.

```
# hastart
```

2 Verify that the CP server service group (CPSSG) is online.

```
# hagrp -state CPSSG
```

Output similar to the following appears:

```
# Group Attribute  System                     Value
  CPSSG State      cps1.example.com  |ONLINE|
```

Configuring CP server using response files

You can configure a CP server using a generated responsefile.

On a single node VCS cluster:

- Run the `installer` command with the responsefile option to configure the CP server on a single node VCS cluster.

```
# /opt/VRTS/install/installer -responsefile '/tmp/sample1.res'
```

On a SFHA cluster:

- Run the `installer` command with the responsefile option to configure the CP server on a SFHA cluster.

```
# /opt/VRTS/install/installer -responsefile '/tmp/sample1.res'
```

Response file variables to configure CP server

Table 2-3 describes the response file variables to configure CP server.

Table 2-3 describes response file variables to configure CP server

Variable	List or Scalar	Description
CFG{opt}{configcps}	Scalar	This variable performs CP server configuration task
CFG{cps_singlenode_config}	Scalar	This variable describes if the CP server will be configured on a singlenode VCS cluster
CFG{cps_sfha_config}	Scalar	This variable describes if the CP server will be configured on a SFHA cluster
CFG{cps_unconfig}	Scalar	This variable describes if the CP server will be unconfigured
CFG{cpsname}	Scalar	This variable describes the name of the CP server
CFG{cps_db_dir}	Scalar	This variable describes the absolute path of CP server database
CFG{cps_reuse_cred}	Scalar	This variable describes if reusing the existing credentials for the CP server
CFG{cps_https_vips}	List	This variable describes the virtual IP addresses for the CP server configured for HTTPS-based communication
CFG{cps_https_ports}	List	This variable describes the port number for the virtual IP addresses for the CP server configured for HTTPS-based communication
CFG{cps_nic_list}{cpsvip<n>}	List	This variable describes the NICs of the systems for the virtual IP address
CFG{cps_netmasks}	List	This variable describes the netmasks for the virtual IP addresses
CFG{cps_prefix_length}	List	This variable describes the prefix length for the virtual IP addresses
CFG{cps_network_hosts}{cpsnic<n>}	List	This variable describes the network hosts for the NIC resource
CFG{cps_vip2nicres_map}{<vip>}	Scalar	This variable describes the NIC resource to associate with the virtual IP address

Table 2-3 describes response file variables to configure CP server *(continued)*

Variable	List or Scalar	Description
CFG{cps_diskgroup}	Scalar	This variable describes the disk group for the CP server database
CFG{cps_volume}	Scalar	This variable describes the volume for the CP server database
CFG{cps_newdg_disks}	List	This variable describes the disks to be used to create a new disk group for the CP server database
CFG{cps_newvol_volsize}	Scalar	This variable describes the volume size to create a new volume for the CP server database
CFG{cps_delete_database}	Scalar	This variable describes if deleting the database of the CP server during the unconfiguration
CFG{cps_delete_config_log}	Scalar	This variable describes if deleting the config files and log files of the CP server during the unconfiguration
CFG{cps_reconfig}	Scalar	This variable defines if the CP server will be reconfigured

Sample response file for configuring the CP server on single node VCS cluster

Review the response file variables and their definitions.

See Table 2-3 on page 46.

```
# Configuration Values:
#
our %CFG;

$CFG{cps_db_dir}="/etc/VRTScps/db";
$CFG{cps_https_ports}=[ 443 ];
$CFG{cps_https_vips}=[ "192.168.59.77" ];
$CFG{cps_netmasks}=[ "255.255.248.0" ];
$CFG{cps_network_hosts}{cpsnic1}=
[ "10.200.117.70" ];
$CFG{cps_nic_list}{cpsvip1}=[ "en0" ];
$CFG{cps_singlenode_config}=1;
$CFG{cps_vip2nicres_map}{"192.168.59.77"}=1;
$CFG{cpsname}="cps1";
$CFG{opt}{configcps}=1;
$CFG{opt}{configure}=1;
$CFG{opt}{noipc}=1;
$CFG{opt}{redirect}=1;
$CFG{prod}="AVAILABILITY742";
$CFG{systems}=[ "linux1" ];
$CFG{vcs_clusterid}=23172;
$CFG{vcs_clustername}="clus72";

1;
```

Sample response file for configuring the CP server on SFHA cluster

Review the response file variables and their definitions.

See Table 2-3 on page 46.

```
#
# Configuration Values:
#
our %CFG;

$CFG{cps_db_dir}="/cpsdb";
$CFG{cps_diskgroup}="cps_dg1";
$CFG{cps_https_ports}=[ qw(50006 50007) ];
$CFG{cps_https_vips}=[ qw(10.198.90.6 10.198.90.7) ];
$CFG{cps_netmasks}=[ qw(255.255.248.0 255.255.248.0 255.255.248.0) ];
$CFG{cps_network_hosts}{cpsnic1}=[ qw(10.198.88.18) ];
$CFG{cps_network_hosts}{cpsnic2}=[ qw(10.198.88.18) ];
$CFG{cps_newdg_disks}=[ qw(emc_clariion0_249) ];
$CFG{cps_newvol_volsize}=10;
$CFG{cps_nic_list}{cpsvip1}=[ qw(eth0 eth0) ];
$CFG{cps_sfha_config}=1;
$CFG{cps_vip2nicres_map}{"10.198.90.6"}=1;
$CFG{cps_volume}="volcps";
$CFG{cpsname}="cps1";
```

```
$CFG{opt}{configcps}=1;
$CFG{opt}{configure}=1;
$CFG{opt}{noipc}=1;

$CFG{prod}="ENTERPRISE742";

$CFG{systems}=[ qw(sys1 sys2) ];
$CFG{vcs_clusterid}=49604;
$CFG{vcs_clustername}="sfha2233";

1;
```

Verifying the CP server configuration

Perform the following steps to verify the CP server configuration.

To verify the CP server configuration

1. Verify that the following configuration files are updated with the information you provided during the CP server configuration process:

   - /etc/vxcps.conf (CP server configuration file)
   - /etc/VRTSvcs/conf/config/main.cf (VCS configuration file)
   - /etc/VRTScps/db (default location for CP server database for a single-node cluster)
   - /cps_db (default location for CP server database for a multi-node cluster)

2. Run the `cpsadm` command to check if the vxcpserv process is listening on the configured Virtual IP.

   If the application cluster is configured for HTTPS-based communication, no need to provide the port number assigned for HTTP communication.

   ```
   # cpsadm -s cp_server -a ping_cps
   ```

   where *cp_server* is the virtual IP address or the virtual hostname of the CP server.

Configuring VCS

This chapter includes the following topics:

Overview of tasks to configure VCS using the product installer

Table 3-1 lists the tasks that are involved in configuring VCS using the script-based installer.

Table 3-1 Tasks to configure VCS using the script-based installer

Task	Reference
Start the software configuration	See “Starting the software configuration” on page 51.
Specify the systems where you want to configure VCS	See “Specifying systems for configuration” on page 52.
Configure the basic cluster	See “Configuring the cluster name” on page 53. See “Configuring private heartbeat links” on page 53.
Configure virtual IP address of the cluster (optional)	See “Configuring the virtual IP of the cluster” on page 60.
Configure the cluster in secure mode (optional)	See “Configuring VCS in secure mode” on page 62.
Add VCS users (required if you did not configure the cluster in secure mode)	See “Adding VCS users” on page 69.
Configure SMTP email notification (optional)	See “Configuring SMTP email notification” on page 70.
Configure SNMP email notification (optional)	See “Configuring SNMP trap notification” on page 71.
Configure global clusters (optional)	See “Configuring global clusters” on page 73.
Complete the software configuration	See “Completing the VCS configuration” on page 73.

Starting the software configuration

You can configure VCS using the product installer.

Note: If you want to reconfigure VCS, before you start the installer you must stop all the resources that are under VCS control using the `hastop` command or the `hagrp -offline` command.

To configure VCS using the product installer

1. Confirm that you are logged in as a superuser.
2. Start the configuration using the installer.

   ```
   # /opt/VRTS/install/installer -configure
   ```

   The installer starts the product installation program with a copyright message and specifies the directory where the logs are created.
3. Select the component to configure.
4. Continue with the configuration procedure by responding to the installer questions.

Specifying systems for configuration

The installer prompts for the system names on which you want to configure VCS. The installer performs an initial check on the systems that you specify.

To specify system names for configuration

1. Enter the names of the systems where you want to configure VCS.

   ```
   Enter the operating_system system names separated
   by spaces:  [q,?] (sys1) sys1 sys2
   ```

2. Review the output as the installer verifies the systems you specify.

   The installer does the following tasks:

   - Checks that the local node running the installer can communicate with remote nodes
     If the installer finds ssh binaries, it confirms that ssh can operate without requests for passwords or passphrases. If ssh binaries cannot communicate with remote nodes, the installer tries rsh binaries. And if both ssh and rsh binaries fail, the installer prompts to help the user to setup ssh or rsh binaries.
   - Makes sure that the systems are running with the supported operating system
   - Checks whether Veritas InfoScale Enterprise is installed

- Exits if Veritas InfoScale Enterprise7.4.2 is not installed

3 Review the installer output about the I/O fencing configuration and confirm whether you want to configure fencing in enabled mode.

```
Do you want to configure I/O Fencing in enabled mode? [y,n,q,?] (y)
```

See “ About planning to configure I/O fencing” on page 20.

Configuring the cluster name

Enter the cluster information when the installer prompts you.

To configure the cluster

1 Review the configuration instructions that the installer presents.

2 Enter a unique cluster name.

```
Enter the unique cluster name: [q,?] clus1
```

Configuring private heartbeat links

You now configure the private heartbeat links that LLT uses.

VCS provides the option to use LLT over Ethernet or LLT over UDP (User Datagram Protocol) or LLT over RDMA, or LLT over TCP. Veritas recommends that you configure heartbeat links that use LLT over Ethernet or LLT over RDMA for high performance, unless hardware requirements force you to use LLT over UDP. If you want to configure LLT over UDP, make sure you meet the prerequisites.

You must not configure LLT heartbeat using the links that are part of aggregated links. For example, link1, link2 can be aggregated to create an aggregated link, aggr1. You can use aggr1 as a heartbeat link, but you must not use either link1 or link2 as heartbeat links.

See “Using the UDP layer for LLT” on page 290.

See “Using the TCP layer for LLT” on page 307.

See “Using LLT over RDMA: supported use cases ” on page 327.

The following procedure helps you configure LLT heartbeat links.

To configure private heartbeat links

1 Choose one of the following options at the installer prompt based on whether you want to configure LLT over Ethernet or LLT over UDP or LLT over TCP or LLT over RDMA.

- Option 1: Configure the heartbeat links using LLT over Ethernet (answer installer questions)

  Enter the heartbeat link details at the installer prompt to configure LLT over Ethernet.

  Skip to step 2.

- Option 2: Configure the heartbeat links using LLT over UDP (answer installer questions)

  Make sure that each NIC you want to use as heartbeat link has an IP address configured. Enter the heartbeat link details at the installer prompt to configure LLT over UDP. If you had not already configured IP addresses to the NICs, the installer provides you an option to detect the IP address for a given NIC.

  Note: Ensure that the interface that is used by the LLT links does not have any other IP in the same subnet were the LLT links are configured. Otherwise, the cluster may behave unpredictably.

  Skip to step 3.

- Option 3: Configure the heartbeat links using LLT over TCP (answer installer questions)

  Make sure that the NIC you want to use as heartbeat link has an IP address configured. Enter the heartbeat link details at the installer prompt to configure LLT over TCP. If you had not already configured IP addresses to the NICs, the installer provides you an option to detect the IP address for a given NIC.

  Skip to step 4.

- Option 4: Configure the heartbeat links using LLT over RDMA (answer installer questions)

  Make sure that each RDMA enabled NIC (RNIC) you want to use as heartbeat link has an IP address configured. Enter the heartbeat link details at the installer prompt to configure LLT over RDMA. If you had not already configured IP addresses to the RNICs, the installer provides you an option to detect the IP address for a given RNIC.

  Skip to step 5.

- Option 5: Automatically detect configuration for LLT over Ethernet

  Allow the installer to automatically detect the heartbeat link details to configure LLT over Ethernet. The installer tries to detect all connected links between all systems.

  Make sure that you activated the NICs for the installer to be able to detect and automatically configure the heartbeat links.

  Skip to step 8.

Note: Option 5 is not available when the configuration is a single node configuration.

2 If you chose option 1, enter the network interface card details for the private heartbeat links.

The installer discovers and lists the network interface cards.

You must not enter the network interface card that is used for the public network (typically eth0.)

```
Enter the NIC for the first private heartbeat link on sys1:
[b,q,?] eth1
eth1 has an IP address configured on it. It could be a
public NIC on sys1.
Are you sure you want to use eth1 for the first private
heartbeat link? [y,n,q,b,?] (n) y
Would you like to configure a second private heartbeat link?
[y,n,q,b,?] (y)
Enter the NIC for the second private heartbeat link on sys1:
[b,q,?] eth2
eth2 has an IP address configured on it. It could be a
public NIC on sys1.
Are you sure you want to use eth2 for the second private
heartbeat link? [y,n,q,b,?] (n) y
Would you like to configure a third private heartbeat link?
[y,n,q,b,?](n)

Do you want to configure an additional low priority heartbeat
link? [y,n,q,b,?] (n)
```

3 If you chose option 2, enter the NIC details for the private heartbeat links. This step uses examples such as *private_NIC1* or *private_NIC2* to refer to the available names of the NICs.

```
Enter the NIC for the first private heartbeat link on sys1: [b,q,?]
private_NIC1
Some configured IP addresses have been found on
the NIC private_NIC1 in sys1,
Do you want to choose one for the first private heartbeat link? [y,n,q,?]
Please select one IP address:
     1)  192.168.0.1/24
     2)  192.168.1.233/24
     b)  Back to previous menu

Please select one IP address: [1-2,b,q,?] (1)
Enter the UDP port for the first private heartbeat link on sys1:
[b,q,?] (50000)

Enter the NIC for the second private heartbeat link on sys1: [b,q,?]
private_NIC2
Some configured IP addresses have been found on the
NIC private_NIC2 in sys1,
Do you want to choose one for the second
private heartbeat link? [y,n,q,?] (y)
Please select one IP address:
     1)  192.168.1.1/24
     2)  192.168.2.233/24
     b)  Back to previous menu

Please select one IP address: [1-2,b,q,?] (1) 1
Enter the UDP port for the second private heartbeat link on sys1:
[b,q,?] (50001)

Would you like to configure a third private heartbeat
link? [y,n,q,b,?] (n)

Do you want to configure an additional low-priority heartbeat
link? [y,n,q,b,?] (n) y

Enter the NIC for the low-priority heartbeat link on sys1: [b,q,?]
private_NIC0
Some configured IP addresses have been found on
```

```
the NIC private_NIC0 in sys1,
Do you want to choose one for the low-priority
heartbeat link? [y,n,q,?] (y)
Please select one IP address:
     1)  10.200.59.233/22
     2)  192.168.3.1/22
     b)  Back to previous menu

Please select one IP address: [1-2,b,q,?] (1) 2
Enter the UDP port for the low-priority heartbeat link on sys1:
[b,q,?] (50010)
```

4 If you chose option 3, enter the NIC details for the private heartbeat link.

This step uses an example such as *private_NIC1* to refer to the available name of the NIC.

```
Enter the NIC for the private heartbeat link on sys1: [b,q,?] (eth1)
private_NIC1
Some configured IP addresses have been found on
the NIC private_NIC1 in sys1,
Do you want to choose one for the private
heartbeat link? [y,n,q,?] (y) y
Please select one IP address:
1) 192.168.1.1/24
2) 192.168.2.1/24
b) Back to previous menu

Please select one IP address: [1-2,b,q,?] (1)
Enter the TCP port for the first private heartbeat link on sys1:
[b,q,?] (50000)
```

5 If you chose option 4, choose the interconnect type to configure RDMA.

```
1)  Converged Ethernet (RoCE)
2)  InfiniBand
b)  Back to previous menu

Choose the RDMA interconnect type [1-2,b,q,?] (1) 2
```

The system displays the details such as the required OS files, drivers required for RDMA , and the IP addresses for the NICs.

A sample output of the IP addresses assigned to the RDMA enabled NICs using InfiniBand network. Note that with RoCE, the RDMA NIC values are represented as eth0, eth1, and so on.

```
System        RDMA NIC          IP Address
=================================================================
sys1           ib0               192.168.0.1
sys1           ib1               192.168.3.1
sys2           ib0               192.168.0.2
sys2           ib1               192.168.3.2
```

6. If you chose option 4, enter the NIC details for the private heartbeat links. This step uses RDMA over an InfiniBand network. With RoCE as the interconnect type, RDMA NIC is represented as Ethernet (eth).

```
Enter the NIC for the first private heartbeat
link (RDMA) on sys1: [b,q,?] <ib0>

Do you want to use address 192.168.0.1 for the
first private heartbeat link on sys1: [y,n,q,b,?] (y)

Enter the port for the first private heartbeat
link (RDMA) on sys1: [b,q,?] (50000) ?

Would you like to configure a second private
heartbeat link? [y,n,q,b,?] (y)
Enter the NIC for the second private heartbeat link (RDMA) on sys1:
[b,q,?] (ib1)

Do you want to use the address 192.168.3.1 for the second
private heartbeat link on sys1: [y,n,q,b,?] (y)
Enter the port for the second private heartbeat link (RDMA) on sys1:
[b,q,?] (50001)

Do you want to configure an additional low-priority heartbeat link?
[y,n,q,b,?] (n)
```

7. Choose whether to use the same NIC details to configure private heartbeat links on other systems.

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

If you want to use the NIC details that you entered for sys1, make sure the same NICs are available on each system. Then, enter **y** at the prompt.

If the NIC device names are different on some of the systems, enter **n**. Provide the NIC details for each system as the program prompts.

For LLT over UDP and LLT over RDMA, if you want to use the same NICs on other systems, you must enter unique IP addresses on each NIC for other systems.

8. If you chose option 5, the installer detects NICs on each system and network links, and sets link priority.

   If the installer fails to detect heartbeat links or fails to find any high-priority links, then choose option 1 or option 2 to manually configure the heartbeat links.

   See step 2 for option 1, or step 3 for option 2, or step 4 for option 3, or step5 for option 4

9. Enter a unique cluster ID:

   ```
   Enter a unique cluster ID number between 0-65535: [b,q,?] (60842)
   ```

   The cluster cannot be configured if the cluster ID 60842 is in use by another cluster. Installer performs a check to determine if the cluster ID is duplicate. The check takes less than a minute to complete.

   ```
   Would you like to check if the cluster ID is in use by another
   cluster? [y,n,q] (y)
   ```

10. Verify and confirm the information that the installer summarizes.

Configuring the virtual IP of the cluster

You can configure the virtual IP of the cluster to use to connect from the Cluster Manager (Java Console), Veritas InfoScale Operations Manager, or to specify in the RemoteGroup resource.

See the *Cluster Server Administrator's Guide* for information on the Cluster Manager.

See the *Cluster Server Bundled Agents Reference Guide* for information on the RemoteGroup agent.

To configure the virtual IP of the cluster

1. Review the required information to configure the virtual IP of the cluster.

2. When the system prompts whether you want to configure the virtual IP, enter `y`.

3. Confirm whether you want to use the discovered public NIC on the first system.

   Do one of the following:

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

```
Active NIC devices discovered on sys1: eth0
Enter the NIC for Virtual IP of the Cluster to use on sys1:
[b,q,?](eth0)
```

4 Confirm whether you want to use the same public NIC on all nodes.

Do one of the following:

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

```
Is eth0 to be the public NIC used by all systems
[y,n,q,b,?] (y)
```

5 Enter the virtual IP address for the cluster.

You can enter either an IPv4 address or an IPv6 address.

For IPv4:

- Enter the virtual IP address.

  ```
  Enter the Virtual IP address for the Cluster:
  [b,q,?] 192.168.1.16
  ```

- Confirm the default netmask or enter another one:

  ```
  Enter the netmask for IP 192.168.1.16: [b,q,?]
  (255.255.240.0)
  ```

- Verify and confirm the Cluster Virtual IP information.

  ```
  Cluster Virtual IP verification:

        NIC: eth0
        IP: 192.168.1.16
        Netmask: 255.255.240.0

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

For IPv6

- Enter the virtual IP address.

  ```
  Enter the Virtual IP address for the Cluster:
  [b,q,?] 2001:454e:205a:110:203:baff:feee:10
  ```

- Enter the prefix for the virtual IPv6 address you provided. For example:

  ```
  Enter the Prefix for IP
  2001:454e:205a:110:203:baff:feee:10: [b,q,?] 64
  ```

- Verify and confirm the Cluster Virtual IP information.

  ```
  Cluster Virtual IP verification:

        NIC: eth0
        IP: 2001:454e:205a:110:203:baff:feee:10
        Prefix: 64

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

If you want to set up trust relationships for your secure cluster, refer to the following topics:

See "Setting up trust relationships for your VCS cluster" on page 63.

See "Configuring a secure cluster node by node" on page 64.

Configuring VCS in secure mode

Configuring VCS in secure mode ensures that all the communication between the systems is encrypted and users are verified against security credentials. VCS user names and passwords are not used when a cluster is running in secure mode.

To configure VCS in secure mode

1. To install and configure VCS in secure mode, run the command:

   ```
   # ./installer -security
   ```

2. The installer displays the following question before the installer stops the product processes:

   - Do you want to grant read access to everyone? [y,n,q,?]
     - To grant read access to all authenticated users, type **y**.
     - To grant usergroup specific permissions, type **n**.

- Do you want to provide any usergroups that you would like to grant read access?[y,n,q,?]
  - To specify usergroups and grant them read access, type **y**
  - To grant read access only to root users, type **n**. The installer grants read access read access to the root users.
- Enter the usergroup names separated by spaces that you would like to grant read access. If you would like to grant read access to a usergroup on a specific node, enter like 'usrgrp1@node1', and if you would like to grant read access to usergroup on any cluster node, enter like 'usrgrp1'. If some usergroups are not created yet, create the usergroups after configuration if needed. [b]

3 To verify the cluster is in secure mode after configuration, run the command:

```
# haclus -value SecureClus
```

The command returns 1 if cluster is in secure mode, else returns 0.

Setting up trust relationships for your VCS cluster

If you need to use an external authentication broker for authenticating VCS users, you must set up a trust relationship between VCS and the broker. For example, if Veritas InfoScale Operations Manager is your external authentication broker, the trust relationship ensures that VCS accepts the credentials that VOM issues.

Perform the following steps to set up a trust relationship between your VCS cluster and a broker.

To set up a trust relationship

1 Ensure that you are logged in as superuser on one of the nodes in the cluster.

2 Enter the following command:

```
# /opt/VRTS/install/installer  -securitytrust
```

The installer specifies the location of the log files. It then lists the cluster information such as cluster name, cluster ID, node names, and service groups.

3 When the installer prompts you for the broker information, specify the IP address, port number, and the data directory for which you want to establish trust relationship with the broker.

```
Input the broker name of IP address: 15.193.97.204
```

```
Input the broker port: (14545)
```

Specify a port number on which broker is running or press Enter to accept the default port.

```
Input the data directory to setup trust with: (/var/VRTSvcs/
vcsauth/data/HAD)
```

Specify a valid data directory or press Enter to accept the default directory.

4 The installer performs one of the following actions:

- If you specified a valid directory, the installer prompts for a confirmation.

  ```
  Are you sure that you want to setup trust for the VCS cluster
  with the broker 15.193.97.204 and port 14545? [y,n,q] y
  ```

  The installer sets up trust relationship with the broker for all nodes in the cluster and displays a confirmation.

  ```
  Setup trust with broker 15.193.97.204 on cluster node1
  ........Done
  ```

  ```
  Setup trust with broker 15.193.97.204 on cluster node2
  ........Done
  ```

  The installer specifies the location of the log files, summary file, and response file and exits.

- If you entered incorrect details for broker IP address, port number, or directory name, the installer displays an error. It specifies the location of the log files, summary file, and response file and exits.

Configuring a secure cluster node by node

For environments that do not support passwordless ssh or passwordless rsh, you cannot use the `-security` option to enable secure mode for your cluster. Instead, you can use the `-securityonenode` option to configure a secure cluster node by node. Moreover, to enable security in fips mode, use the `-fips` option together with `-securityonenode`.

Table 3-2 lists the tasks that you must perform to configure a secure cluster.

Table 3-2 Configuring a secure cluster node by node

Task	Reference
Configure security on one node	See "Configuring the first node" on page 65.
Configure security on the remaining nodes	See "Configuring the remaining nodes" on page 66.
Complete the manual configuration steps	See "Completing the secure cluster configuration" on page 66.

Configuring the first node

Perform the following steps on one node in your cluster.

To configure security on the first node

1. Ensure that you are logged in as superuser.
2. Enter the following command:

```
# /opt/VRTS/install/installer -securityonenode
```

The installer lists information about the cluster, nodes, and service groups. If VCS is not configured or if VCS is not running on all nodes of the cluster, the installer prompts whether you want to continue configuring security. It then prompts you for the node that you want to configure.

```
VCS is not running on all systems in this cluster. All VCS systems
must be in RUNNING state. Do you want to continue? [y,n,q] (n) y

1) Perform security configuration on first node and export
security configuration files.

2) Perform security configuration on remaining nodes with
security configuration files.

Select the option you would like to perform [1-2,q.?] 1
```

Warning: All VCS configurations about cluster users are deleted when you configure the first node. You can use the `/opt/VRTSvcs/bin/hauser` command to create cluster users manually.

3. The installer completes the secure configuration on the node. It specifies the location of the security configuration files and prompts you to copy these files to the other nodes in the cluster. The installer also specifies the location of log files, summary file, and response file.
4. Copy the security configuration files from the location specified by the installer to temporary directories on the other nodes in the cluster.

Configuring the remaining nodes

On each of the remaining nodes in the cluster, perform the following steps.

To configure security on each remaining node

1. Ensure that you are logged in as superuser.
2. Enter the following command:

   ```
   # /opt/VRTS/install/installer -securityonenode
   ```

   The installer lists information about the cluster, nodes, and service groups. If VCS is not configured or if VCS is not running on all nodes of the cluster, the installer prompts whether you want to continue configuring security. It then prompts you for the node that you want to configure. Enter **2**.

   ```
   VCS is not running on all systems in this cluster. All VCS systems
   must be in RUNNING state. Do you want to continue? [y,n,q] (n) y

   1) Perform security configuration on first node and export
   security configuration files.

   2) Perform security configuration on remaining nodes with
   security configuration files.

   Select the option you would like to perform [1-2,q.?]  2
   Enter the security conf file directory: [b]
   ```

   The installer completes the secure configuration on the node. It specifies the location of log files, summary file, and response file.

Completing the secure cluster configuration

Perform the following manual steps to complete the configuration.

To complete the secure cluster configuration

1. On the first node, freeze all service groups except the ClusterService service group.

   ```
   # /opt/VRTSvcs/bin/haconf -makerw
   ```

   ```
   # /opt/VRTSvcs/bin/hagrp -list Frozen=0
   ```

   ```
   # /opt/VRTSvcs/bin/hagrp -freeze groupname -persistent
   ```

   ```
   # /opt/VRTSvcs/bin/haconf -dump -makero
   ```

2. On the first node, stop the VCS engine.

   ```
   # /opt/VRTSvcs/bin/hastop -all -force
   ```

3. On all nodes, stop the CmdServer.

   ```
   # systemctl stop CmdServer
   ```

4. To grant access to all users, add or modify `SecureClus=1` and `DefaultGuestAccess=1` in the cluster definition.

   For example:

   To grant read access to everyone:

   ```
   Cluster clus1 (
   SecureClus=1
   DefaultGuestAccess=1
   )
   ```

   Or

   To grant access to only root:

   ```
   Cluster clus1 (
   SecureClus=1
   )
   ```

   Or

   To grant read access to specific user groups, add or modify SecureClus=1 and GuestGroups={} to the cluster definition.

   For example:

   ```
   cluster clus1 (
   SecureClus=1
   GuestGroups={staff, guest}
   ```

5. Modify `/etc/VRTSvcs/conf/config/main.cf` file on the first node, and add `-secure` to the WAC application definition if GCO is configured.

   For example:

   ```
   Application wac (
                  StartProgram = "/opt/VRTSvcs/bin/wacstart -secure"
                  StopProgram = "/opt/VRTSvcs/bin/wacstop"
                  MonitorProcesses = {"/opt/VRTSvcs/bin/wac -secure"}
                  RestartLimit = 3
                  )
   ```

6. On all nodes, create the `/etc/VRTSvcs/conf/config/.secure` file.

   ```
   # touch /etc/VRTSvcs/conf/config/.secure
   ```

7. On the first node, start VCS. Then start VCS on the remaining nodes.

   ```
   # /opt/VRTSvcs/bin/hastart
   ```

8. On all nodes, start CmdServer.

   ```
   # systemctl start CmdServer
   ```

9. On the first node, unfreeze the service groups.

   ```
   # /opt/VRTSvcs/bin/haconf -makerw
   ```

   ```
   # /opt/VRTSvcs/bin/hagrp -list Frozen=1
   ```

   ```
   # /opt/VRTSvcs/bin/hagrp -unfreeze groupname -persistent
   ```

   ```
   # /opt/VRTSvcs/bin/haconf -dump -makero
   ```

Adding VCS users

If you have enabled a secure VCS cluster, you do not need to add VCS users now. Otherwise, on systems operating under an English locale, you can add VCS users at this time.

To add VCS users

1. Review the required information to add VCS users.
2. Reset the password for the Admin user, if necessary.

   ```
   Do you wish to accept the default cluster credentials of
   'admin/password'? [y,n,q] (y) n
   Enter the user name: [b,q,?] (admin)
   Enter the password:
   Enter again:
   ```

   The password is encrypted using the standard AES-256 algorithm.

3. To add a user, enter **y** at the prompt.

   ```
   Do you want to add another user to the cluster? [y,n,q] (y)
   ```

4. Enter the user’s name, password, and level of privileges.

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

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

   The password is encrypted using the standard AES-256 algorithm.

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

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

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

Configuring SMTP email notification

You can choose to configure VCS to send event notifications to SMTP email services. You need to provide the SMTP server name and email addresses of people to be notified. Note that you can also configure the notification after installation.

Refer to the *Cluster Server Administrator's Guide* for more information.

To configure SMTP email notification

1. Review the required information to configure the SMTP email notification.
2. Specify whether you want to configure the SMTP notification.

   If you do not want to configure the SMTP notification, you can skip to the next configuration option.

   See “Configuring SNMP trap notification” on page 71.

3. Provide information to configure SMTP notification.

   Provide the following information:

   - Enter the SMTP server’s host name.

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

   - Enter the email address of each recipient.

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

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

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

4. Add more SMTP recipients, if necessary.

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

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

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

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

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

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

5. Verify and confirm the SMTP notification information.

   ```
   SMTP Address: smtp.example.com
   Recipient: ozzie@example.com receives email for Warning or
   higher events
   Recipient: harriet@example.com receives email for Error or
   higher events

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

Configuring SNMP trap notification

You can choose to configure VCS to send event notifications to SNMP management consoles. You need to provide the SNMP management console name to be notified and message severity levels.

Note that you can also configure the notification after installation.

Refer to the *Cluster Server Administrator's Guide* for more information.

To configure the SNMP trap notification

1. Review the required information to configure the SNMP notification feature of VCS.
2. Specify whether you want to configure the SNMP notification.

   If you skip this option and if you had installed a valid HA/DR license, the installer presents you with an option to configure this cluster as global cluster. If you did not install an HA/DR license, the installer proceeds to configure VCS based on the configuration details you provided.

   See "Configuring global clusters" on page 73.
3. Provide information to configure SNMP trap notification.

   Provide the following information:

   - Enter the SNMP trap daemon port.

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

   - Enter the SNMP console system name.

     ```
     Enter the SNMP console system name: [b,q,?] sys5
     ```

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

     ```
     Enter the minimum severity of events for which SNMP traps
     should be sent to sys5 [I=Information, W=Warning, E=Error,
     S=SevereError]: [b,q,?] E
     ```

4. Add more SNMP consoles, if necessary.

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

     ```
     Would you like to add another SNMP console? [y,n,q,b] (n) y
     Enter the SNMP console system name: [b,q,?] sys4
     Enter the minimum severity of events for which SNMP traps
     should be sent to sys4 [I=Information, W=Warning,
     E=Error, S=SevereError]: [b,q,?] S
     ```

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

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

5 Verify and confirm the SNMP notification information.

```
SNMP Port: 162
Console: sys5 receives SNMP traps for Error or
higher events
Console: sys4 receives SNMP traps for SevereError or
higher events

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

Configuring global clusters

You can configure global clusters to link clusters at separate locations and enable wide-area failover and disaster recovery. The installer adds basic global cluster information to the VCS configuration file. You must perform additional configuration tasks to set up a global cluster.

See the *Cluster Server Administrator's Guide* for instructions to set up VCS global clusters.

Note: If you installed a HA/DR license to set up replicated data cluster or campus cluster, skip this installer option.

To configure the global cluster option

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

2 Specify whether you want to configure the global cluster option.

If you skip this option, the installer proceeds to configure VCS based on the configuration details you provided.

3 Provide information to configure this cluster as global cluster.

The installer prompts you for a NIC, a virtual IP address, and value for the netmask.

You can also enter an IPv6 address as a virtual IP address.

Completing the VCS configuration

After you enter the VCS configuration information, the installer prompts to stop the VCS processes to complete the configuration process. The installer continues to create configuration files and copies them to each system. The installer also configures a cluster UUID value for the cluster at the end of the configuration. After the installer successfully configures VCS, it restarts VCS and its related processes.

To complete the VCS configuration

1. If prompted, press Enter at the following prompt.

   ```
   Do you want to stop InfoScale Enterprise  processes now? [y,n,q,?] (y)
   ```

2. Review the output as the installer stops various processes and performs the configuration. The installer then restarts VCS and its related processes.

3. Enter y at the prompt to send the installation information to Veritas.

   ```
   Would you like to send the information about this installation
   to us  to help improve installation in the future?
   [y,n,q,?] (y) y
   ```

4. After the installer configures VCS successfully, note the location of summary, log, and response files that installer creates.

   The files provide the useful information that can assist you with the configuration and can also assist future configurations.

summary file	Describes the cluster and its configured resources.
log file	Details the entire configuration.
response file	Contains the configuration information that can be used to perform secure or unattended installations on other systems. See “Configuring VCS using response files” on page 109.

Verifying the NIC configuration

The installer verifies on all the nodes if all NICs have PERSISTENT_NAME set correctly.

If the persistent interface names are not configured correctly for the network devices, the installer displays the following messages:

```
PERSISTENT_NAME is not set for all the NICs.
You need to set them manually before the next reboot.
```

Set the PERSISTENT_NAME for all the NICs.

Warning: If the installer finds the network interface name to be different from the name in the configuration file, then the installer exits.

About Veritas License Audit Tool

Veritas License Audit Tool by Veritas intelligently scans your organization’s network and gives you a comprehensive report of all the Veritas product licenses used at your organization. This robust tool allows your organization to see all the current Veritas products installed your systems. This helps your organization in the following:

- License and maintenance renewal of Veritas products
- Contract renegotiations of Veritas Products
- Re-harvesting and reuse of Veritas Products

Veritas License Audit Tool's robust reporting framework enables you to capture information such as Product name, Product Version, Licensing key, License type, Operating System, Operating System Version and CPU Name.

To download the Veritas License Audit Tool and its Installation and User Guide, click the following link:

https://sort.veritas.com/public/utilities/infoscale/latool/linux/LATool-rhel7.tar

Verifying and updating licenses on the system

After you install Cluster Server you can verify and manage the licenses using the vxlicrep program.

See “Checking licensing information on the system” on page 75.

See “Replacing a VCS keyless license with another keyless license” on page 77.

Checking licensing information on the system

You can use the vxlicrep program to display information about the licenses on a system.

To check licensing information

1. Navigate to the `/sbin` folder containing the `vxlicrep` program and enter:

   ```
   # vxlicrep
   ```

2. Review the following output to determine the following information:

   - The license key

- The type of license

Note: Note that the **VXKEYLESS** parameter is displayed in the output of a keyless license and that it is set to **ENABLED**.

- The product for which it applies

Example: Excerpt of output for a keyless license

```
License Key                     = xxxxxxxxxxx.slf
Product Name                    = VCS
Point Product                   = YES

Features :=
COUNT                           = 1
COUNT POLICY                    = max:100%
GLOBAL CLUSTER OPTION           = Enabled
LICENSE METER                   = PER-CORE
LICENSE TYPE                    = PERPETUAL
MODE                            = VCS
PLATFORM                        = UNIX
PLATFORM POLICY                 = Hard
PRODUCT EDITION                 = AVAILABILITY
PRODUCT ID                      = 115
TIER                            = Tier 3
TIER POLICY                     = Soft
VERSION                         = 7.4.1
VXKEYLESS                       = ENABLED
```

Example: Excerpt of output for a permanent license

```
License Key                     = xxxxxxxxxxx.slf
Product Name                    = VCS
Point Product                   = YES

Features :=
COUNT                           = 10
COUNT POLICY                    = max:100%
GLOBAL CLUSTER OPTION           = Enabled
LICENSE METER                   = PER-CORE
LICENSE TYPE                    = PERPETUAL
MODE                            = VCS
PLATFORM                        = UNIX
PLATFORM POLICY                 = Hard
```

```
PRODUCT EDITION                = AVAILABILITY
PRODUCT ID                     = 115
SVC POLICY                     = Soft
TIER                           = Tier 3
TIER POLICY                    = Soft
VERSION                        = 7.4.1
```

Replacing a VCS keyless license with another keyless license

You can use the `./installer -license` command or the `vxkeyless` command to replace a VCS keyless license with another keyless license on each node.

See “Replacing a VCS keyless license with a permanent license” on page 77.

To update product licenses using the installer command

1. On any one node, enter the following command:

   ```
   # ./installer -license
   ```

2. At the prompt, enter your keyless license text string.

To update product licenses using the vxkeyless command

◆ On each node, enter the keyless license text string using the command:

```
# vxkeyless set <keyless license text-string>
```

Example:

```
# vxkeyless set ENTERPRISE
```

Replacing a VCS keyless license with a permanent license

Within 60 days of enabling the VCS keyless license, you must replace it with a permanent license using the `vxlicinstupgrade` program.

To update product licenses using the vxlicinstupgrade command

1 Make sure you have permissions to log in as root on each of the nodes in the cluster.

2 Enter the permanent license key using the following command on each node:

```
# vxlicinstupgrade -k <key file path>
```

Note: The license key file must not be saved in the root directory (`/`) or the default license directory on the local host (`/etc/vx/licenses/lic`). You can save the license key file inside any other directory on the local host.

---

3 Make sure keyless licenses are replaced on all cluster nodes before starting VCS.

```
# vxlicrep
```

To update product licenses using the installer command

1 On any one node, enter the following command:

```
#./installer -license
```

2 At the prompt, enter your permanent license key file.

Chapter 4

Configuring VCS clusters for data integrity

This chapter includes the following topics:

Setting up disk-based I/O fencing using installer

You can configure I/O fencing using the `-fencing` option of the installer.

Initializing disks as VxVM disks

Perform the following procedure to initialize disks as VxVM disks.

To initialize disks as VxVM disks

1. List the new external disks or the LUNs as recognized by the operating system. On each node, enter:

   ```
   # fdisk -l
   ```

2. To initialize the disks as VxVM disks, use one of the following methods:

   - Use the interactive vxdiskadm utility to initialize the disks as VxVM disks. For more information, see the *Storage Foundation Administrator's Guide*.

- Use the `vxdisksetup` command to initialize a disk as a VxVM disk.

  ```
  # vxdisksetup -i device_name
  ```

  The example specifies the CDS format:

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

  Repeat this command for each disk you intend to use as a coordinator disk.

Configuring disk-based I/O fencing using installer

Note: The installer stops and starts VCS to complete I/O fencing configuration. Make sure to unfreeze any frozen VCS service groups in the cluster for the installer to successfully stop VCS.

To set up disk-based I/O fencing using the installer

1 Start the installer with `-fencing` option.

```
# /opt/VRTS/install/installer  -fencing
```

The installer starts with a copyright message and verifies the cluster information.

Note the location of log files which you can access in the event of any problem with the configuration process.

2 Enter the host name of one of the systems in the cluster.

3 Confirm that you want to proceed with the I/O fencing configuration at the prompt.

The program checks that the local node running the script can communicate with remote nodes and checks whether VCS 7.4.2 is configured properly.

4 Review the I/O fencing configuration options that the program presents. Type **2** to configure disk-based I/O fencing.

```
1. Configure Coordination Point client based fencing
2. Configure disk based fencing
3. Configure majority based fencing
4. Configure fencing in disabled mode
Select the fencing mechanism to be configured in this
Application Cluster [1-4,q.?] 2
```

5 Review the output as the configuration program checks whether VxVM is already started and is running.

- If the check fails, configure and enable VxVM before you repeat this procedure.
- If the check passes, then the program prompts you for the coordinator disk group information.

6 Choose whether to use an existing disk group or create a new disk group to configure as the coordinator disk group.

The program lists the available disk group names and provides an option to create a new disk group. Perform one of the following:

- To use an existing disk group, enter the number corresponding to the disk group at the prompt.
  The program verifies whether the disk group you chose has an odd number of disks and that the disk group has a minimum of three disks.
- To create a new disk group, perform the following steps:
  - Enter the number corresponding to the **Create a new disk group** option.
    The program lists the available disks that are in the CDS disk format in the cluster and asks you to choose an odd number of disks with at least three disks to be used as coordinator disks.
    Veritas recommends that you use three disks as coordination points for disk-based I/O fencing.
  - If the available VxVM CDS disks are less than the required, installer asks whether you want to initialize more disks as VxVM disks. Choose the disks you want to initialize as VxVM disks and then use them to create new disk group.
  - Enter the numbers corresponding to the disks that you want to use as coordinator disks.
  - Enter the disk group name.

7 Verify that the coordinator disks you chose meet the I/O fencing requirements.

You must verify that the disks are SCSI-3 PR compatible using the vxfentsthdw utility and then return to this configuration program.

See “Checking shared disks for I/O fencing” on page 84.

8 After you confirm the requirements, the program creates the coordinator disk group with the information you provided.

9 Verify and confirm the I/O fencing configuration information that the installer summarizes.

10 Review the output as the configuration program does the following:

- Stops VCS and I/O fencing on each node.
- Configures disk-based I/O fencing and starts the I/O fencing process.
- Updates the VCS configuration file main.cf if necessary.
- Copies the /etc/vxfenmode file to a date and time suffixed file /etc/vxfenmode-*date-time*. This backup file is useful if any future fencing configuration fails.
- Updates the I/O fencing configuration file /etc/vxfenmode.
- Starts VCS on each node to make sure that the VCS is cleanly configured to use the I/O fencing feature.

11 Review the output as the configuration program displays the location of the log files, the summary files, and the response files.

12 Configure the Coordination Point Agent.

```
Do you want to configure Coordination Point Agent on
the client cluster? [y,n,q] (y)
```

13 Enter a name for the service group for the Coordination Point Agent.

```
Enter a non-existing name for the service group for
Coordination Point Agent: [b] (vxfen) vxfen
```

14 Set the level two monitor frequency.

```
Do you want to set LevelTwoMonitorFreq? [y,n,q] (y)
```

15 Decide the value of the level two monitor frequency.

```
Enter the value of the LevelTwoMonitorFreq attribute: [b,q,?] (5)
```

Installer adds Coordination Point Agent and updates the main configuration file.

16 Enable auto refresh of coordination points.

```
Do you want to enable auto refresh of coordination points
if registration keys are missing
on any of them? [y,n,q,b,?]  (n)
```

See “Configuring CoordPoint agent to monitor coordination points” on page 173.

Refreshing keys or registrations on the existing coordination points for disk-based fencing using the installer

You must refresh registrations on the coordination points in the following scenarios:

- When the CoordPoint agent notifies VCS about the loss of registration on any of the existing coordination points.
- A planned refresh of registrations on coordination points when the cluster is online without having an application downtime on the cluster.

Registration loss may happen because of an accidental array restart, corruption of keys, or some other reason. If the coordination points lose the registrations of the cluster nodes, the cluster may panic when a network partition occurs.

Warning: Refreshing keys might cause the cluster to panic if a node leaves membership before the coordination points refresh is complete.

To refresh registrations on existing coordination points for disk-based I/O fencing using the installer

1 Start the installer with the `-fencing` option.

```
# /opt/VRTS/install/installer  -fencing
```

The installer starts with a copyright message and verifies the cluster information.

Note down the location of log files that you can access if there is a problem with the configuration process.

2 Confirm that you want to proceed with the I/O fencing configuration at the prompt.

The program checks that the local node running the script can communicate with the remote nodes and checks whether VCS 7.4.2 is configured properly.

3 Review the I/O fencing configuration options that the program presents. Type the number corresponding to refresh registrations or keys on the existing coordination points.

```
Select the fencing mechanism to be configured in this
Application Cluster [1-6,q]
```

4 Ensure that the disk group constitution that is used by the fencing module contains the same disks that are currently used as coordination disks.

5. Verify the coordination points.

   ```
   For example,
   Disk Group: fendg
   Fencing disk policy: dmp
   Fencing disks:
    emc_clariion0_62
    emc_clariion0_65
    emc_clariion0_66
   ```

   Is this information correct? [y,n,q] **(y)**.

   ```
   Successfully completed the vxfenswap operation
   ```

   The keys on the coordination disks are refreshed.

6. Do you want to send the information about this installation to us to help improve installation in the future? [y,n,q,?] **(y)**.

7. Do you want to view the summary file? [y,n,q] **(n)**.

Checking shared disks for I/O fencing

Make sure that the shared storage you set up while preparing to configure VCS meets the I/O fencing requirements. You can test the shared disks using the vxfentsthdw utility. The two nodes must have `ssh` (default) or `rsh` communication. 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. This command option verifies that the same serial number for the LUN is returned on all paths to the LUN.

Make sure to test the disks that serve as coordinator disks.

The vxfentsthdw utility has additional options suitable for testing many disks. Review the options for testing the disk groups (`-g`) and the disks that are listed in a file (`-f`). You can also test disks without destroying data using the `-r` option.

See the *Cluster Server Administrator's Guide*.

Checking that disks support SCSI-3 involves the following tasks:

- Verifying the Array Support Library (ASL)
  See "Verifying Array Support Library (ASL)" on page 85.

- Verifying that nodes have access to the same disk
  See "Verifying that the nodes have access to the same disk" on page 86.

- Testing the shared disks for SCSI-3
  See "Testing the disks using vxfentsthdw utility" on page 86.

Verifying Array Support Library (ASL)

Make sure that the Array Support Library (ASL) for the array that you add is installed.

To verify Array Support Library (ASL)

1 If the Array Support Library (ASL) for the array that you add is not installed, obtain and install it on each node before proceeding.

  The ASL for the supported storage device that you add is available from the disk array vendor or Veritas technical support.

2 Verify that the ASL for the disk array is installed on each of the nodes. Run the following command on each node and examine the output to verify the installation of ASL.

  The following output is a sample:

```
# vxddladm listsupport all


LIBNAME             VID                 PID
=============================================================
libvxhitachi.so     HITACHI             DF350, DF400, DF400F,
                                        DF500, DF500F
libvxxp1281024.so   HP                  All
libvxxp12k.so       HP                  All
libvxddns2a.so      DDN                 S2A 9550, S2A 9900,
                                        S2A 9700
libvxpurple.so      SUN                 T300
libvxxiotechE5k.so  XIOTECH             ISE1400
libvxcopan.so       COPANSYS            8814, 8818
libvxibmds8k.so     IBM                 2107
```

3 Scan all disk drives and their attributes, update the VxVM device list, and reconfigure DMP with the new devices. Type:

```
# vxdisk scandisks
```

  See the Veritas Volume Manager documentation for details on how to add and configure disks.

Verifying that the nodes have access to the same disk

Before you test the disks that you plan to use as shared data storage or as coordinator disks using the vxfentsthdw utility, you must verify that the systems see the same disk.

To verify that the nodes have access to the same disk

1. Verify the connection of the shared storage for data to two of the nodes on which you installed Veritas InfoScale Enterprise.
2. Ensure that both nodes are connected to the same disk during the testing. Use the vxfenadm command to verify the disk serial number.

   ```
   # vxfenadm -i diskpath
   ```

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

   For example, an EMC disk is accessible by the /dev/sdx path on node A and the /dev/sdy path on node B.

   From node A, enter:

   ```
   # vxfenadm -i /dev/sdx

   SCSI ID=>Host: 2 Channel: 0 Id: 0 Lun: E

   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/sdy path.

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

   ```
   SCSI ID=>Host: 2 Channel: 0 Id: 0 Lun: E

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

Testing the disks using vxfentsthdw utility

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

If the utility does not show a message that states a disk is ready, the verification has failed. Failure of verification can be the result of an improperly configured disk array. The failure can also be due to a bad disk.

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

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

For more information on how to replace coordinator disks, refer to the *Cluster Server Administrator's Guide*.

To test the disks using vxfentsthdw utility

1. Make sure system-to-system communication functions properly.

   See “About configuring secure shell or remote shell communication modes before installing products” on page 346.

2. From one node, start the utility.

3. The script warns that the tests overwrite data on the disks. After you review the overview and the warning, confirm to continue the process and enter the node names.

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

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

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

4. Review the output as the utility performs the checks and reports its activities.

5 If a disk is ready for I/O fencing on each node, the utility reports success for each node. For example, the utility displays the following message for the node sys1.

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

ALL tests on the disk /dev/sdx have PASSED
The disk is now ready to be configured for I/O fencing on node
sys1
```

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

Note: Only dmp disk devices can be used as coordinator disks.

Setting up server-based I/O fencing using installer

You can configure server-based I/O fencing for the VCS cluster using the installer.

With server-based fencing, you can have the coordination points in your configuration as follows:

- Combination of CP servers and SCSI-3 compliant coordinator disks
- CP servers only
  Veritas also supports server-based fencing with a single highly available CP server that acts as a single coordination point.

See “ About planning to configure I/O fencing” on page 20.

See “Recommended CP server configurations” on page 26.

This section covers the following example procedures:

Mix of CP servers and coordinator disks	See “To configure server-based fencing for the VCS cluster (one CP server and two coordinator disks)” on page 88.
Single CP server	See “To configure server-based fencing for the VCS cluster” on page 93.

To configure server-based fencing for the VCS cluster (one CP server and two coordinator disks)

1 Depending on the server-based configuration model in your setup, make sure of the following:

- CP servers are configured and are reachable from the VCS cluster. The VCS cluster is also referred to as the application cluster or the client cluster.

  See "Setting up the CP server" on page 28.
- The coordination disks are verified for SCSI3-PR compliance.

  See "Checking shared disks for I/O fencing" on page 84.

2 Start the installer with the `-fencing` option.

```
# /opt/VRTS/install/installer  -fencing
```

The installer starts with a copyright message and verifies the cluster information.

Note the location of log files which you can access in the event of any problem with the configuration process.

3 Confirm that you want to proceed with the I/O fencing configuration at the prompt.

The program checks that the local node running the script can communicate with remote nodes and checks whether VCS 7.4.2 is configured properly.

4 Review the I/O fencing configuration options that the program presents. Type **1** to configure server-based I/O fencing.

```
Select the fencing mechanism to be configured in this
Application Cluster [1-3,b,q] 1
```

5 Make sure that the storage supports SCSI3-PR, and answer y at the following prompt.

```
Does your storage environment support SCSI3 PR? [y,n,q] (y)
```

6 Provide the following details about the coordination points at the installer prompt:

- Enter the total number of coordination points including both servers and disks. This number should be at least 3.

```
Enter the total number of co-ordination points including both
Coordination Point servers and disks: [b] (3)
```

- Enter the total number of coordinator disks among the coordination points.

```
Enter the total number of disks among these:
[b] (0) 2
```

7 Provide the following CP server details at the installer prompt:

- Enter the total number of virtual IP addresses or the total number of fully qualified host names for each of the CP servers.

  ```
  How many IP addresses would you like to use to communicate
   to Coordination Point Server #1?: [b,q,?] (1) 1
  ```

- Enter the virtual IP addresses or the fully qualified host name for each of the CP servers. The installer assumes these values to be identical as viewed from all the application cluster nodes.

  ```
  Enter the Virtual IP address or fully qualified host name #1
  for the HTTPS Coordination Point Server #1:
  [b] 10.209.80.197
  ```

  The installer prompts for this information for the number of virtual IP addresses you want to configure for each CP server.

- Enter the port that the CP server would be listening on.

  ```
  Enter the port that the coordination point server  10.209.80.197
  would be listening on or accept the default port
  suggested: [b] (443)
  ```

8 Provide the following coordinator disks-related details at the installer prompt:

- Choose the coordinator disks from the list of available disks that the installer displays. Ensure that the disk you choose is available from all the VCS (application cluster) nodes.
  The number of times that the installer asks you to choose the disks depends on the information that you provided in step 6. For example, if you had chosen to configure two coordinator disks, the installer asks you to choose the first disk and then the second disk:

  ```
  Select disk number 1 for co-ordination point

  1) sdx
  2) sdy
  3) sdz

  Please enter a valid disk which is available from all the
  cluster nodes for co-ordination point [1-3,q] 1
  ```

- If you have not already checked the disks for SCSI-3 PR compliance in step 1, check the disks now.

The installer displays a message that recommends you to verify the disks in another window and then return to this configuration procedure.

Press Enter to continue, and confirm your disk selection at the installer prompt.

- Enter a disk group name for the coordinator disks or accept the default.

```
Enter the disk group name for coordinating disk(s):
[b] (vxfencoorddg)
```

9 Verify and confirm the coordination points information for the fencing configuration.

For example:

```
Total number of coordination points being used: 3
Coordination Point Server ([VIP or FQHN]:Port):
    1. 10.209.80.197 ([10.209.80.197]:443)
SCSI-3 disks:
    1. sdx
    2. sdy
Disk Group name for the disks in customized fencing: vxfencoorddg
Disk policy used for customized fencing: dmp
```

The installer initializes the disks and the disk group and deports the disk group on the VCS (application cluster) node.

10 Verify and confirm the I/O fencing configuration information.

```
CPS Admin utility location: /opt/VRTScps/bin/cpsadm
Cluster ID: 2122
Cluster Name: clus1
UUID for the above cluster: {ae5e589a-1dd1-11b2-dd44-00144f79240c}
```

11. Review the output as the installer updates the application cluster information on each of the CP servers to ensure connectivity between them. The installer then populates the `/etc/vxfenmode` file with the appropriate details in each of the application cluster nodes.

```
Updating client cluster information on Coordination Point Server 10.209.80.197

Adding the client cluster to the Coordination Point Server 10.209.80.197 .......... Done

Registering client node sys1 with Coordination Point Server 10.209.80.197...... Done
Adding CPClient user for communicating to Coordination Point Server 10.209.80.197 .... Done
Adding cluster clus1 to the CPClient user on Coordination Point Server 10.209.80.197 .. Done

Registering client node sys2 with Coordination Point Server 10.209.80.197 ..... Done
Adding CPClient user for communicating to Coordination Point Server 10.209.80.197 .... Done
Adding cluster clus1 to the CPClient user on Coordination Point Server 10.209.80.197 ..Done

Updating /etc/vxfenmode file on sys1 .................................. Done
Updating /etc/vxfenmode file on sys2 ......... ........................ Done
```

    See “About I/O fencing configuration files” on page 278.

12. Review the output as the installer stops and restarts the VCS and the fencing processes on each application cluster node, and completes the I/O fencing configuration.

13. Configure the CP agent on the VCS (application cluster). The Coordination Point Agent monitors the registrations on the coordination points.

    ```
    Do you want to configure Coordination Point Agent on
    the client cluster? [y,n,q] (y)

    Enter a non-existing name for the service group for
    Coordination Point Agent: [b] (vxfen)
    ```

14 Additionally the coordination point agent can also monitor changes to the Coordinator Disk Group constitution such as a disk being accidently deleted from the Coordinator Disk Group. The frequency of this detailed monitoring can be tuned with the LevelTwoMonitorFreq attribute. For example, if you set this attribute to 5, the agent will monitor the Coordinator Disk Group constitution every five monitor cycles.

   Note that for the LevelTwoMonitorFreq attribute to be applicable there must be disks as part of the Coordinator Disk Group.

   ```
   Enter the value of the LevelTwoMonitorFreq attribute: (5)
   ```

15 Enable auto refresh of coordination points.

   ```
   Do you want to enable auto refresh of coordination points
   if registration keys are missing
   on any of them? [y,n,q,b,?]  (n)
   ```

16 Note the location of the configuration log files, summary files, and response files that the installer displays for later use.

17 Verify the fencing configuration using:

   ```
   # vxfenadm -d
   ```

18 Verify the list of coordination points.

   ```
   # vxfenconfig -l
   ```

To configure server-based fencing for the VCS cluster

1 Make sure that the CP server is configured and is reachable from the VCS cluster. The VCS cluster is also referred to as the application cluster or the client cluster.

2 See "Setting up the CP server" on page 28.

3 Start the installer with `-fencing` option.

   ```
   # /opt/VRTS/install/installer  -fencing
   ```

   The installer starts with a copyright message and verifies the cluster information.

   Note the location of log files which you can access in the event of any problem with the configuration process.

4 Confirm that you want to proceed with the I/O fencing configuration at the prompt.

The program checks that the local node running the script can communicate with remote nodes and checks whether VCS 7.4.2 is configured properly.

5 Review the I/O fencing configuration options that the program presents. Type **1** to configure server-based I/O fencing.

```
Select the fencing mechanism to be configured in this
Application Cluster [1-7,q] 1
```

6 Make sure that the storage supports SCSI3-PR, and answer y at the following prompt.

```
Does your storage environment support SCSI3 PR? [y,n,q] (y)
```

7 Enter the total number of coordination points as **1**.

```
Enter the total number of co-ordination points including both
Coordination Point servers and disks: [b] (3) 1
```

Read the installer warning carefully before you proceed with the configuration.

8 Provide the following CP server details at the installer prompt:

- Enter the total number of virtual IP addresses or the total number of fully qualified host names for each of the CP servers.

  ```
  How many IP addresses would you like to use to communicate
  to Coordination Point Server #1? [b,q,?] (1) 1
  ```

- Enter the virtual IP address or the fully qualified host name for the CP server. The installer assumes these values to be identical as viewed from all the application cluster nodes.

  ```
  Enter the Virtual IP address or fully qualified host name
  #1 for the Coordination Point Server #1:
  [b] 10.209.80.197
  ```

  The installer prompts for this information for the number of virtual IP addresses you want to configure for each CP server.

- Enter the port that the CP server would be listening on.

  ```
  Enter the port in the range [49152, 65535] which the
  Coordination Point Server 10.209.80.197
  ```

```
  would be listening on or simply accept the default
  port suggested: [b] (443)
```

9. Verify and confirm the coordination points information for the fencing configuration.

   For example:

   ```
   Total number of coordination points being used: 1
   Coordination Point Server ([VIP or FQHN]:Port):
       1. 10.209.80.197 ([10.209.80.197]:443)
   ```

10. Verify and confirm the I/O fencing configuration information.

    ```
    CPS Admin utility location: /opt/VRTScps/bin/cpsadm
    Cluster ID: 2122
    Cluster Name: clus1
    UUID for the above cluster: {ae5e589a-1dd1-11b2-dd44-00144f79240c}
    ```

11. Review the output as the installer updates the application cluster information on each of the CP servers to ensure connectivity between them. The installer then populates the `/etc/vxfenmode` file with the appropriate details in each of the application cluster nodes.

    The installer also populates the `/etc/vxfenmode` file with the entry `single_cp=1` for such single CP server fencing configuration.

```
Updating client cluster information on Coordination Point Server 10.209.80.197

Adding the client cluster to the Coordination Point Server 10.209.80.197 .......... Done

Registering client node sys1 with Coordination Point Server 10.209.80.197...... Done
Adding CPClient user for communicating to Coordination Point Server 10.209.80.197 .... Done
Adding cluster clus1 to the CPClient user on Coordination Point Server 10.209.80.197 .. Done

Registering client node sys2 with Coordination Point Server 10.209.80.197 ..... Done
Adding CPClient user for communicating to Coordination Point Server 10.209.80.197 .... Done
Adding cluster clus1 to the CPClient user on Coordination Point Server 10.209.80.197 .. Done

Updating /etc/vxfenmode file on sys1 .................................. Done
Updating /etc/vxfenmode file on sys2 ......... ........................ Done
```

See “About I/O fencing configuration files” on page 278.

12 Review the output as the installer stops and restarts the VCS and the fencing processes on each application cluster node, and completes the I/O fencing configuration.

13 Configure the CP agent on the VCS (application cluster).

```
Do you want to configure Coordination Point Agent on the
client cluster? [y,n,q] (y)

Enter a non-existing name for the service group for
Coordination Point Agent: [b] (vxfen)
```

14 Enable auto refresh of coordination points.

```
Do you want to enable auto refresh of coordination points
if registration keys are missing
on any of them? [y,n,q,b,?]  (n)
```

15 Note the location of the configuration log files, summary files, and response files that the installer displays for later use.

Refreshing keys or registrations on the existing coordination points for server-based fencing using the installer

You must refresh registrations on the coordination points in the following scenarios:

- When the CoordPoint agent notifies VCS about the loss of registration on any of the existing coordination points.
- A planned refresh of registrations on coordination points when the cluster is online without having an application downtime on the cluster.

Registration loss might occur because of an accidental array restart, corruption of keys, or some other reason. If the coordination points lose registrations of the cluster nodes, the cluster might panic when a network partition occurs.

Warning: Refreshing keys might cause the cluster to panic if a node leaves membership before the coordination points refresh is complete.

To refresh registrations on existing coordination points for server-based I/O fencing using the installer

1. Start the installer with the `-fencing` option.

   ```
   # /opt/VRTS/install/installer  -fencing
   ```

   The installer starts with a copyright message and verifies the cluster information.

   Note the location of log files that you can access if there is a problem with the configuration process.

2. Confirm that you want to proceed with the I/O fencing configuration at the prompt.

   The program checks that the local node running the script can communicate with the remote nodes and checks whether VCS 7.4.2 is configured properly.

3. Review the I/O fencing configuration options that the program presents. Type the number corresponding to the option that suggests to refresh registrations or keys on the existing coordination points.

   ```
   Select the fencing mechanism to be configured in this
   Application Cluster [1-7,q] 6
   ```

4. Ensure that the `/etc/vxfentab` file contains the same coordination point servers that are currently used by the fencing module.

   Also, ensure that the disk group mentioned in the `/etc/vxfendg` file contains the same disks that are currently used by the fencing module as coordination disks.

5. Verify the coordination points.

   ```
   For example,
   Total number of coordination points being used: 3
   Coordination Point Server ([VIP or FQHN]:Port):
         1. 10.198.94.146 ([10.198.94.146]:443)
         2. 10.198.94.144 ([10.198.94.144]:443)
   SCSI-3 disks:
         1. emc_clariion0_61
   Disk Group name for the disks in customized fencing: vxfencoorddg
   Disk policy used for customized fencing: dmp
   ```

6. Is this information correct? [y,n,q] **(y)**

   ```
   Updating client cluster information on Coordination Point Server
    IPaddress

   Successfully completed the vxfenswap operation
   ```

   The keys on the coordination disks are refreshed.

7. Do you want to send the information about this installation to us to help improve installation in the future? [y,n,q,?] **(y)**.

8. Do you want to view the summary file? [y,n,q] **(n)**.

Setting the order of existing coordination points for server-based fencing using the installer

This section describes the reasons, benefits, considerations, and the procedure to set the order of the existing coordination points for server-based fencing.

About deciding the order of existing coordination points

You can decide the order in which coordination points can participate in a race during a network partition. In a network partition scenario, I/O fencing attempts to contact coordination points for membership arbitration based on the order that is set in the `vxfentab` file.

When I/O fencing is not able to connect to the first coordination point in the sequence it goes to the second coordination point and so on. To avoid a cluster panic, the surviving subcluster must win majority of the coordination points. So, the order must begin with the coordination point that has the best chance to win the race and must end with the coordination point that has the least chance to win the race.

For fencing configurations that use a mix of coordination point servers and coordination disks, you can specify either coordination point servers before coordination disks or disks before servers.

Note: Disk-based fencing does not support setting the order of existing coordination points.

Considerations to decide the order of coordination points

- Choose the coordination points based on their chances to gain membership on the cluster during the race and hence gain control over a network partition. In effect, you have the ability to save a partition.

- First in the order must be the coordination point that has the best chance to win the race. The next coordination point you list in the order must have relatively lesser chance to win the race. Complete the order such that the last coordination point has the least chance to win the race.

Setting the order of existing coordination points using the installer

To set the order of existing coordination points

1 Start the installer with `-fencing` option.

```
# /opt/VRTS/install/installer  -fencing
```

The installer starts with a copyright message and verifies the cluster information.

Note the location of log files that you can access if there is a problem with the configuration process.

2 Confirm that you want to proceed with the I/O fencing configuration at the prompt.

The program checks that the local node running the script can communicate with remote nodes and checks whether VCS 7.4.2 is configured properly.

3 Review the I/O fencing configuration options that the program presents. Type the number corresponding to the option that suggests to set the order of existing coordination points.

For example:

```
Select the fencing mechanism to be configured in this
Application Cluster [1-7,q] 7


Installer will ask the new order of existing coordination points.
Then it will call vxfenswap utility to commit the
coordination points change.
```

Warning: The cluster might panic if a node leaves membership before the coordination points change is complete.

4. Review the current order of coordination points.

```
Current coordination points order:
(Coordination disks/Coordination Point Server)
Example,
1)  /dev/vx/rdmp/emc_clariion0_65,/dev/vx/rdmp/emc_clariion0_66,
/dev/vx/rdmp/emc_clariion0_62
2)  [10.198.94.144]:443
3)  [10.198.94.146]:443
b)  Back to previous menu
```

5. Enter the new order of the coordination points by the numbers and separate the order by space [1-3,b,q] **3 1 2**.

```
New coordination points order:
(Coordination disks/Coordination Point Server)
Example,
1) [10.198.94.146]:443
2) /dev/vx/rdmp/emc_clariion0_65,/dev/vx/rdmp/emc_clariion0_66,
/dev/vx/rdmp/emc_clariion0_62
3) [10.198.94.144]:443
```

6. Is this information correct? [y,n,q] **(y)**.

```
Preparing vxfenmode.test file on all systems...
Running vxfenswap...
Successfully completed the vxfenswap operation
```

7. Do you want to send the information about this installation to us to help improve installation in the future? [y,n,q,?] **(y)**.

8. Do you want to view the summary file? [y,n,q] **(n)**.

9 Verify that the value of `vxfen_honor_cp_order` specified in the `/etc/vxfenmode` file is set to **1**.

```
For example,
vxfen_mode=customized
vxfen_mechanism=cps
port=443
scsi3_disk_policy=dmp
cps1=[10.198.94.146]
vxfendg=vxfencoorddg
cps2=[10.198.94.144]
vxfen_honor_cp_order=1
```

10 Verify that the coordination point order is updated in the output of the `vxfenconfig -l` command.

```
For example,
I/O Fencing Configuration Information:
======================================

single_cp=0
[10.198.94.146]:443 {e7823b24-1dd1-11b2-8814-2299557f1dc0}
/dev/vx/rdmp/emc_clariion0_65  60060160A38B1600386FD87CA8FDDD11
/dev/vx/rdmp/emc_clariion0_66  60060160A38B1600396FD87CA8FDDD11
/dev/vx/rdmp/emc_clariion0_62  60060160A38B16005AA00372A8FDDD11
[10.198.94.144]:443 {01f18460-1dd2-11b2-b818-659cbc6eb360}
```

Setting up non-SCSI-3 I/O fencing in virtual environments using installer

If you have installed Veritas InfoScale Enterprise in virtual environments that do not support SCSI-3 PR-compliant storage, you can configure non-SCSI-3 fencing.

To configure I/O fencing using the installer in a non-SCSI-3 PR-compliant setup

1 Start the installer with `-fencing` option.

```
# /opt/VRTS/install/installer  -fencing
```

The installer starts with a copyright message and verifies the cluster information.

2 Confirm that you want to proceed with the I/O fencing configuration at the prompt.

The program checks that the local node running the script can communicate with remote nodes and checks whether VCS 7.4.2 is configured properly.

3 For server-based fencing, review the I/O fencing configuration options that the program presents. Type **1** to configure server-based I/O fencing.

```
Select the fencing mechanism to be configured in this
Application Cluster
[1-7,q] 1
```

4 Enter **n** to confirm that your storage environment does not support SCSI-3 PR.

```
Does your storage environment support SCSI3 PR?
[y,n,q] (y) n
```

5 Confirm that you want to proceed with the non-SCSI-3 I/O fencing configuration at the prompt.

6 For server-based fencing, enter the number of CP server coordination points you want to use in your setup.

7 For server-based fencing, enter the following details for each CP server:

- Enter the virtual IP address or the fully qualified host name.
- Enter the port address on which the CP server listens for connections.
  The default value is 443. You can enter a different port address. Valid values are between 49152 and 65535.

The installer assumes that these values are identical from the view of the VCS cluster nodes that host the applications for high availability.

8 For server-based fencing, verify and confirm the CP server information that you provided.

9 Verify and confirm the VCS cluster configuration information.

Review the output as the installer performs the following tasks:

- Updates the CP server configuration files on each CP server with the following details for only server-based fencing, :
  - Registers each node of the VCS cluster with the CP server.
  - Adds CP server user to the CP server.
  - Adds VCS cluster to the CP server user.
- Updates the following configuration files on each node of the VCS cluster
  - `/etc/vxfenmode` file
  - `/etc/vxenviron` file
  - `/etc/sysconfig/vxfen` file
  - `/etc/llttab` file
  - `/etc/vxfentab` (only for server-based fencing)

10 Review the output as the installer stops VCS on each node, starts I/O fencing on each node, updates the VCS configuration file main.cf, and restarts VCS with non-SCSI-3 fencing.

For server-based fencing, confirm to configure the CP agent on the VCS cluster.

11 Confirm whether you want to send the installation information to us.

12 After the installer configures I/O fencing successfully, note the location of summary, log, and response files that installer creates.

The files provide useful information which can assist you with the configuration, and can also assist future configurations.

Setting up majority-based I/O fencing using installer

You can configure majority-based fencing for the cluster using the installer .

Perform the following steps to confgure majority-based I/O fencing

1. Start the installer with the -fencing option.

   ```
   # /opt/VRTS/install/installer  -fencing
   ```

   Where *version* is the specific release version. The installer starts with a copyright message and verifies the cluster information.

   Note: Make a note of the log file location which you can access in the event of any issues with the configuration process.

2. Confirm that you want to proceed with the I/O fencing configuration at the prompt. The program checks that the local node running the script can communicate with remote nodes and checks whether VCS is configured properly.

3. Review the I/O fencing configuration options that the program presents. Type **3** to configure majority-based I/O fencing.

   ```
   Select the fencing mechanism to be configured in this
   Application Cluster [1-7,b,q] 3
   ```

   Note: The installer will ask the following question. Does your storage environment support SCSI3 PR? [y,n,q,?] Input 'y' if your storage environment supports SCSI3 PR. Other alternative will result in installer configuring non-SCSI3 fencing(NSF).

4. The installer then populates the /etc/vxfenmode file with the appropriate details in each of the application cluster nodes.

   ```
   Updating /etc/vxfenmode file on sys1 .................. Done
   Updating /etc/vxfenmode file on sys2 .................. Done
   ```

5. Review the output as the installer stops and restarts the VCS and the fencing processes on each application cluster node, and completes the I/O fencing configuration.

6. Note the location of the configuration log files, summary files, and response files that the installer displays for later use.

7. Verify the fencing configuration.

   ```
   #  vxfenadm -d
   ```

Enabling or disabling the preferred fencing policy

You can enable or disable the preferred fencing feature for your I/O fencing configuration.

You can enable preferred fencing to use system-based race policy, group-based race policy, or site-based policy. If you disable preferred fencing, the I/O fencing configuration uses the default count-based race policy.

Preferred fencing is not applicable to majority-based I/O fencing.

To enable preferred fencing for the I/O fencing configuration

1. Make sure that the cluster is running with I/O fencing set up.

   ```
   # vxfenadm -d
   ```

2. Make sure that the cluster-level attribute UseFence has the value set to SCSI3.

   ```
   # haclus -value UseFence
   ```

3. To enable system-based race policy, perform the following steps:

   - Make the VCS configuration writable.

     ```
     # haconf -makerw
     ```

   - Set the value of the cluster-level attribute PreferredFencingPolicy as System.

     ```
     # haclus -modify PreferredFencingPolicy System
     ```

   - Set the value of the system-level attribute FencingWeight for each node in the cluster.
     For example, in a two-node cluster, where you want to assign sys1 five times more weight compared to sys2, run the following commands:

     ```
     # hasys -modify sys1 FencingWeight 50
     # hasys -modify sys2 FencingWeight 10
     ```

   - Save the VCS configuration.

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

   - Verify fencing node weights using:

     ```
     # vxfenconfig -a
     ```

4. To enable group-based race policy, perform the following steps:

   - Make the VCS configuration writable.

     ```
     # haconf -makerw
     ```

   - Set the value of the cluster-level attribute PreferredFencingPolicy as Group.

     ```
     # haclus -modify PreferredFencingPolicy Group
     ```

   - Set the value of the group-level attribute Priority for each service group.

     For example, run the following command:

     ```
     # hagrp -modify service_group Priority 1
     ```

     Make sure that you assign a parent service group an equal or lower priority than its child service group. In case the parent and the child service groups are hosted in different subclusters, then the subcluster that hosts the child service group gets higher preference.

   - Save the VCS configuration.

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

5. To enable site-based race policy, perform the following steps:

   - Make the VCS configuration writable.

     ```
     # haconf -makerw
     ```

   - Set the value of the cluster-level attribute PreferredFencingPolicy as Site.

     ```
     # haclus -modify PreferredFencingPolicy Site
     ```

   - Set the value of the site-level attribute Preference for each site.

     ```
     For example,
     # hasite -modify Pune Preference 2
     ```

   - Save the VCS configuration.

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

6. To view the fencing node weights that are currently set in the fencing driver, run the following command:

   ```
   # vxfenconfig -a
   ```

To disable preferred fencing for the I/O fencing configuration

1. Make sure that the cluster is running with I/O fencing set up.

   ```
   # vxfenadm -d
   ```

2. Make sure that the cluster-level attribute UseFence has the value set to SCSI3.

   ```
   # haclus -value UseFence
   ```

3. To disable preferred fencing and use the default race policy, set the value of the cluster-level attribute PreferredFencingPolicy as Disabled.

   ```
   # haconf -makerw
   # haclus -modify PreferredFencingPolicy Disabled
   # haconf -dump -makero
   ```

Section 2

Automated configuration using response files

Performing an automated VCS configuration

This chapter includes the following topics:

Configuring VCS using response files

Typically, you can use the response file that the installer generates after you perform VCS configuration on one cluster to configure VCS on other clusters.

To configure VCS using response files

1. Make sure the Veritas InfoScale Availability or Enterprise RPMs are installed on the systems where you want to configure VCS.
2. Copy the response file to one of the cluster systems where you want to configure VCS.

   See "Sample response file for configuring Cluster Server" on page 122.

3 Edit the values of the response file variables as necessary.

To configure optional features, you must define appropriate values for all the response file variables that are related to the optional feature.

See “Response file variables to configure VCS” on page 110.

4 Start the configuration from the system to which you copied the response file. For example:

```
# /opt/VRTS/install/installer -responsefile
/tmp/response_file
```

Where `/tmp/response_file` is the response file’s full path name.

Response file variables to configure VCS

Table 5-1 lists the response file variables that you can define to configure VCS.

Table 5-1 Response file variables specific to configuring VCS

Variable	List or Scalar	Description
CFG{opt}{configure}	Scalar	Performs the configuration if the RPMs are already installed. (Required) Set the value to 1 to configure VCS.
CFG{accepteula}	Scalar	Specifies whether you agree with EULA.pdf on the media. (Required)
CFG{activecomponent}	List	Defines the component to be configured. The value is VCS742 for VCS. (Required)

CFG{keys}{keyless}
CFG{keys}{license}
	List	`CFG{keys}{keyless}` gives a list of keyless keys to be registered on the system.
		`CFG{keys}{license}` gives a list of user defined keys to be registered on the system.
		(Optional)

Table 5-1 Response file variables specific to configuring VCS *(continued)*

Variable	List or Scalar	Description
CFG{systems}	List	List of systems on which the product is to be configured. (Required)
CFG{prod}	Scalar	Defines the product for operations. The value is AVAILABILITY742 for Veritas InfoScale Availability. (Required)
CFG{opt}{keyfile}	Scalar	Defines the location of an ssh keyfile that is used to communicate with all remote systems. (Optional)
CFG{opt}{rsh}	Scalar	Defines that *rsh* must be used instead of ssh as the communication method between systems. (Optional)
CFG{opt}{logpath}	Scalar	Mentions the location where the log files are to be copied. The default location is /opt/VRTS/install/logs. **Note:** The installer copies the response files and summary files also to the specified *logpath* location. (Optional)
CFG{uploadlogs}	Scalar	Defines a Boolean value 0 or 1. The value 1 indicates that the installation logs are uploaded to the Veritas website. The value 0 indicates that the installation logs are not uploaded to the Veritas website. (Optional)

Table 5-1 Response file variables specific to configuring VCS *(continued)*

Variable	List or Scalar	Description
CFG{edgeserver_host}	String	Defines the edge server to be used. Enter **telemetry.veritas.com** to use the Veritas Cloud Receiver, which is a preconfigured, cloud-based edge server deployed by Veritas. (Required) **Note:** An edge server is used to collect licensing and platform related information from InfoScale products as part of the Veritas Product Improvement Program. The information collected helps identify how customers deploy and use the product, and enables Veritas to manage customer licenses more efficiently.
CFG{edgeserver_port}	Scalar	Define the port number of the edge server. Enter 443, which is the port number used by the Veritas Cloud Receiver. (Required) **Note:** An edge server is used to collect licensing and platform related information from InfoScale products as part of the Veritas Product Improvement Program. The information collected helps identify how customers deploy and use the product, and enables Veritas to manage customer licenses more efficiently.

Note that some optional variables make it necessary to define other optional variables. For example, all the variables that are related to the cluster service group (csgnic, csgvip, and csgnetmask) must be defined if any are defined. The same is true for the SMTP notification (smtpserver, smtprecp, and smtprsev), the SNMP trap notification (snmpport, snmpcons, and snmpcsev), and the Global Cluster Option (gconic, gcovip, and gconetmask).

Table 5-2 lists the response file variables that specify the required information to configure a basic VCS cluster.

Table 5-2 Response file variables specific to configuring a basic VCS cluster

Variable	List or Scalar	Description
CFG{donotreconfigurevcs}	Scalar	Defines if you need to re-configure VCS. (Optional)
CFG{donotreconfigurefencing}	Scalar	Defines if you need to re-configure fencing. (Optional)
CFG{vcs_clusterid}	Scalar	An integer between 0 and 65535 that uniquely identifies the cluster. (Required)
CFG{vcs_clustername}	Scalar	Defines the name of the cluster. (Required)
CFG{vcs_allowcomms}	Scalar	Indicates whether or not to start LLT and GAB when you set up a single-node cluster. The value can be 0 (do not start) or 1 (start). (Required)
CFG{fencingenabled}	Scalar	In a VCS configuration, defines if fencing is enabled. Valid values are 0 or 1. (Required)

Table 5-3 lists the response file variables that specify the required information to configure LLT over Ethernet.

Table 5-3 Response file variables specific to configuring private LLT over Ethernet

Variable	List or Scalar	Description

CFG{vcs_lltlink#}
{"system"}
	Scalar	Defines the NIC to be used for a private heartbeat link on each system. At least two LLT links are required per system (lltlink1 and lltlink2). You can configure up to four LLT links.
		You must enclose the system name within double quotes.
		(Required)

CFG{vcs_lltlinklowpri#}
{"system"}
	Scalar	Defines a low priority heartbeat link. Typically, lltlinklowpri is used on a public network link to provide an additional layer of communication.
		If you use different media speed for the private NICs, you can configure the NICs with lesser speed as low-priority links to enhance LLT performance. For example, lltlinklowpri1, lltlinklowpri2, and so on.
		You must enclose the system name within double quotes.
		(Optional)

Table 5-4 lists the response file variables that specify the required information to configure LLT over UDP.

Table 5-4 Response file variables specific to configuring LLT over UDP

Variable	List or Scalar	Description
CFG{lltoverudp}=1	Scalar	Indicates whether to configure heartbeat link using LLT over UDP. (Required)

Table 5-4 Response file variables specific to configuring LLT over UDP *(continued)*

Variable	List or Scalar	Description

CFG{vcs_udplink<n>_address}
{<sys1>}
	Scalar	Stores the IP address (IPv4 or IPv6) that the heartbeat link uses on node1.
		You can have four heartbeat links and <n> for this response file variable can take values 1 to 4 for the respective heartbeat links.
		(Required)

CFG
{vcs_udplinklowpri<n>_address}
{<sys1>}
	Scalar	Stores the IP address (IPv4 or IPv6) that the low priority heartbeat link uses on node1.
		You can have four low priority heartbeat links and <n> for this response file variable can take values 1 to 4 for the respective low priority heartbeat links.
		(Required)

CFG{vcs_udplink<n>_port}
{<sys1>}
	Scalar	Stores the UDP port (16-bit integer value) that the heartbeat link uses on node1.
		You can have four heartbeat links and <n> for this response file variable can take values 1 to 4 for the respective heartbeat links.
		(Required)

CFG{vcs_udplinklowpri<n>_port}
{<sys1>}
	Scalar	Stores the UDP port (16-bit integer value) that the low priority heartbeat link uses on node1.
		You can have four low priority heartbeat links and <n> for this response file variable can take values 1 to 4 for the respective low priority heartbeat links.
		(Required)

Table 5-4 Response file variables specific to configuring LLT over UDP *(continued)*

Variable	List or Scalar	Description

CFG{vcs_udplink<n>_netmask}
{<sys1>}
	Scalar	Stores the netmask (prefix for IPv6) that the heartbeat link uses on node1.
		You can have four heartbeat links and <n> for this response file variable can take values 1 to 4 for the respective heartbeat links.
		(Required)

CFG {vcs_udplinklowpri<n>_netmask}
{<sys1>}
	Scalar	Stores the netmask (prefix for IPv6) that the low priority heartbeat link uses on node1.
		You can have four low priority heartbeat links and <n> for this response file variable can take values 1 to 4 for the respective low priority heartbeat links.
		(Required)

Table 5-5 lists the response file variables that specify the required information to configure LLT over RDMA.

Table 5-5 Response file variables specific to configuring LLT over RDMA

Variable	List or Scalar	Description
CFG{lltoverrdma}=1	Scalar	Indicates whether to configure heartbeat link using LLT over RDMA. (Required)

CFG{vcs_rdmalink<n>_address}
{<sys1>}
	Scalar	Stores the IP address (IPv4 or IPv6) that the heartbeat link uses on node1.
		You can have four heartbeat links and <n> for this response file variable can take values 1 to 4 for the respective heartbeat links.
		(Required)

Table 5-5 Response file variables specific to configuring LLT over RDMA *(continued)*

Variable	List or Scalar	Description

CFG
{vcs_rdmalinklowpri<n>_address}
{<sys1>}
	Scalar	Stores the IP address (IPv4 or IPv6) that the low priority heartbeat link uses on node1.
		You can have four low priority heartbeat links and <n> for this response file variable can take values 1 to 4 for the respective low priority heartbeat links.
		(Required)

CFG{vcs_rdmalink<n>_port}
{<sys1>}
	Scalar	Stores the RDMA port (16-bit integer value) that the heartbeat link uses on node1.
		You can have four heartbeat links and <n> for this response file variable can take values 1 to 4 for the respective heartbeat links.
		(Required)

CFG{vcs_rdmalinklowpri<n>_port}
{<sys1>}
	Scalar	Stores the RDMA port (16-bit integer value) that the low priority heartbeat link uses on node1.
		You can have four low priority heartbeat links and <n> for this response file variable can take values 1 to 4 for the respective low priority heartbeat links.
		(Required)

CFG{vcs_rdmalink<n>_netmask}
{<sys1>}
	Scalar	Stores the netmask (prefix for IPv6) that the heartbeat link uses on node1.
		You can have four heartbeat links and <n> for this response file variable can take values 1 to 4 for the respective heartbeat links.
		(Required)

Table 5-5 Response file variables specific to configuring LLT over RDMA *(continued)*

Variable	List or Scalar	Description

CFG {vcs_rdmalinklowpri<n>_netmask}
{<sys1>}
	Scalar	Stores the netmask (prefix for IPv6) that the low priority heartbeat link uses on node1.
		You can have four low priority heartbeat links and <n> for this response file variable can take values 1 to 4 for the respective low priority heartbeat links.
		(Required)

Table 5-6 lists the response file variables that specify the required information to configure virtual IP for VCS cluster.

Table 5-6 Response file variables specific to configuring virtual IP for VCS cluster

Variable	List or Scalar	Description

CFG{vcs_csgnic}
{system}
	Scalar	Defines the NIC device to use on a system. You can enter ‘all’ as a system value if the same NIC is used on all systems.
		(Optional)

CFG{vcs_csgvip}	Scalar	Defines the virtual IP address for the cluster. (Optional)
CFG{vcs_csgnetmask}	Scalar	Defines the Netmask of the virtual IP address for the cluster. (Optional)

Table 5-7 lists the response file variables that specify the required information to configure the VCS cluster in secure mode.

Table 5-7 Response file variables specific to configuring VCS cluster in secure mode

Variable	List or Scalar	Description
CFG{vcs_eat_security}	Scalar	Specifies if the cluster is in secure enabled mode or not.
CFG{opt}{securityonenode}	Scalar	Specifies that the securityonenode option is being used.
CFG{securityonenode_menu}	Scalar	Specifies the menu option to choose to configure the secure cluster one at a time. ■ 1—Configure the first node ■ 2—Configure the other node
CFG{secusrgrps}	List	Defines the user groups which get read access to the cluster. List or scalar: list Optional or required: optional
CFG{rootsecusrgrps}	Scalar	Defines the read access to the cluster only for root and other users or user groups which are granted explicit privileges in VCS objects. (Optional)
CFG{security_conf_dir}	Scalar	Specifies the directory where the configuration files are placed.
CFG{opt}{security}	Scalar	Specifies that the security option is being used.
CFG{defaultaccess}	Scalar	Defines if the user chooses to grant read access to everyone. Optional or required: optional
CFG{vcs_eat_security_fips}	Scalar	Specifies that the enabled security is FIPS compliant.

Table 5-8 lists the response file variables that specify the required information to configure VCS users.

Table 5-8 Response file variables specific to configuring VCS users

Variable	List or Scalar	Description
CFG{vcs_userenpw}	List	List of encoded passwords for VCS users The value in the list can be "Administrators Operators Guests" **Note:** The order of the values for the vcs_userenpw list must match the order of the values in the vcs_username list. (Optional)
CFG{vcs_username}	List	List of names of VCS users (Optional)
CFG{vcs_userpriv}	List	List of privileges for VCS users **Note:** The order of the values for the vcs_userpriv list must match the order of the values in the vcs_username list. (Optional)

Table 5-9 lists the response file variables that specify the required information to configure VCS notifications using SMTP.

Table 5-9 Response file variables specific to configuring VCS notifications using SMTP

Variable	List or Scalar	Description
CFG{vcs_smtpserver}	Scalar	Defines the domain-based hostname (example: smtp.example.com) of the SMTP server to be used for web notification. (Optional)
CFG{vcs_smtprecp}	List	List of full email addresses (example: user@example.com) of SMTP recipients. (Optional)

Table 5-9 Response file variables specific to configuring VCS notifications using SMTP *(continued)*

Variable	List or Scalar	Description
CFG{vcs_smtprsev}	List	Defines the minimum severity level of messages (Information, Warning, Error, SevereError) that listed SMTP recipients are to receive. Note that the ordering of severity levels must match that of the addresses of SMTP recipients. (Optional)

Table 5-10 lists the response file variables that specify the required information to configure VCS notifications using SNMP.

Table 5-10 Response file variables specific to configuring VCS notifications using SNMP

Variable	List or Scalar	Description
CFG{vcs_snmpport}	Scalar	Defines the SNMP trap daemon port (default=162). (Optional)
CFG{vcs_snmpcons}	List	List of SNMP console system names (Optional)
CFG{vcs_snmpcsev}	List	Defines the minimum severity level of messages (Information, Warning, Error, SevereError) that listed SNMP consoles are to receive. Note that the ordering of severity levels must match that of the SNMP console system names. (Optional)

Table 5-11 lists the response file variables that specify the required information to configure VCS global clusters.

Table 5-11 Response file variables specific to configuring VCS global clusters

Variable	List or Scalar	Description

CFG{vcs_gconic}
{system}
	Scalar	Defines the NIC for the Virtual IP that the Global Cluster Option uses. You can enter ‘all’ as a system value if the same NIC is used on all systems.
		(Optional)

CFG{vcs_gcovip}	Scalar	Defines the virtual IP address to that the Global Cluster Option uses. (Optional)
CFG{vcs_gconetmask}	Scalar	Defines the Netmask of the virtual IP address that the Global Cluster Option uses. (Optional)

Sample response file for configuring Cluster Server

Review the response file variables and their definitions.

See “Response file variables to configure VCS” on page 110.

```
#
# Configuration Values:
#
our %CFG;

$CFG{opt}{configure}=1;
$CFG{opt}{gco}=1;
$CFG{prod}="Availability742";
$CFG{systems}=[ qw(sys1 sys2) ];
$CFG{vcs_allowcomms}=1;
$CFG{vcs_clusterid}=13221;
$CFG{vcs_clustername}="clus1";
$CFG{vcs_csgnetmask}="255.255.255.0";
$CFG{vcs_csgnic}{all}="eth0";
$CFG{vcs_csgvip}="10.10.12.1";
$CFG{vcs_gconetmask}="255.255.255.0";
```

```
$CFG{vcs_gcovip}="10.10.12.1";
$CFG{activecomponent}=[ qw(VCS742) ]
$CFG{vcs_lltlink1}{sys1}="eth1";
$CFG{vcs_lltlink1}{sys2}="eth1";
$CFG{vcs_lltlink2}{sys1}="eth2";
$CFG{vcs_lltlink2}{sys2}="eth2";

$CFG{vcs_smtprecp}=[ qw(earnie@example.com) ];
$CFG{vcs_smtprsev}=[ qw(SevereError) ];
$CFG{vcs_smtpserver}="smtp.example.com";
$CFG{vcs_snmpcons}=[ qw(neptune) ];
$CFG{vcs_snmpcsev}=[ qw(SevereError) ];
$CFG{vcs_snmpport}=162;
1;
```

Performing an automated I/O fencing configuration using response files

This chapter includes the following topics:

Configuring I/O fencing using response files

Typically, you can use the response file that the installer generates after you perform I/O fencing configuration to configure I/O fencing for VCS.

To configure I/O fencing using response files

1. Make sure that VCS is configured.
2. Based on whether you want to configure disk-based or server-based I/O fencing, make sure you have completed the preparatory tasks.

   See " About planning to configure I/O fencing" on page 20.
3. Copy the response file to one of the cluster systems where you want to configure I/O fencing.

   See "Sample response file for configuring disk-based I/O fencing" on page 128.

   See "Sample response file for configuring server-based I/O fencing" on page 131.

   See "Sample response file for configuring non-SCSI-3 I/O fencing" on page 133.

   See "Sample response file for configuring majority-based I/O fencing" on page 134.
4. Edit the values of the response file variables as necessary.

   See "Response file variables to configure disk-based I/O fencing" on page 125.

   See "Response file variables to configure server-based I/O fencing" on page 129.

   See "Response file variables to configure non-SCSI-3 I/O fencing" on page 132.

   See "Response file variables to configure majority-based I/O fencing" on page 134.
5. Start the configuration from the system to which you copied the response file. For example:

   ```
   # /opt/VRTS/install/installer
   -responsefile /tmp/response_file
   ```

   Where `/tmp/response_file` is the response file's full path name.

Response file variables to configure disk-based I/O fencing

Table 6-1 lists the response file variables that specify the required information to configure disk-based I/O fencing for VCS.

Table 6-1 Response file variables specific to configuring disk-based I/O fencing

Variable	List or Scalar	Description
CFG{opt}{fencing}	Scalar	Performs the I/O fencing configuration. (Required)
CFG{fencing_option}	Scalar	Specifies the I/O fencing configuration mode. ■ 1—Coordination Point Server-based I/O fencing ■ 2—Coordinator disk-based I/O fencing ■ 3—Disabled-based I/O fencing ■ 4—Online fencing migration ■ 5—Refresh keys/registrations on the existing coordination points ■ 6—Change the order of existing coordination points ■ 7—Majority-based fencing (Required)
CFG{fencing_dgname}	Scalar	Specifies the disk group for I/O fencing. (Optional) **Note:** You must define the fencing_dgname variable to use an existing disk group. If you want to create a new disk group, you must use both the fencing_dgname variable and the fencing_newdg_disks variable.
CFG{fencing_newdg_disks}	List	Specifies the disks to use to create a new disk group for I/O fencing. (Optional) **Note:** You must define the fencing_dgname variable to use an existing disk group. If you want to create a new disk group, you must use both the fencing_dgname variable and the fencing_newdg_disks variable.

Table 6-1 Response file variables specific to configuring disk-based I/O fencing *(continued)*

Variable	**List or Scalar**	**Description**
CFG{fencing_cpagent_monitor_freq}	Scalar	Specifies the frequency at which the Coordination Point Agent monitors for any changes to the Coordinator Disk Group constitution. **Note:** Coordination Point Agent can also monitor changes to the Coordinator Disk Group constitution such as a disk being accidently deleted from the Coordinator Disk Group. The frequency of this detailed monitoring can be tuned with the LevelTwoMonitorFreq attribute. For example, if you set this attribute to 5, the agent will monitor the Coordinator Disk Group constitution every five monitor cycles. If LevelTwoMonitorFreq attribute is not set, the agent will not monitor any changes to the Coordinator Disk Group. 0 means not to monitor the Coordinator Disk Group constitution.
CFG {fencing_config_cpagent}	Scalar	Enter '1' or '0' depending upon whether you want to configure the Coordination Point agent using the installer or not. Enter "0" if you do not want to configure the Coordination Point agent using the installer. Enter "1" if you want to use the installer to configure the Coordination Point agent.
CFG {fencing_cpagentgrp}	Scalar	Name of the service group which will have the Coordination Point agent resource as part of it. **Note:** This field is obsolete if the **fencing_config_cpagent** field is given a value of '0'.

Table 6-1 Response file variables specific to configuring disk-based I/O fencing *(continued)*

Variable	List or Scalar	Description
CFG{fencing_auto_refresh_reg}	Scalar	Enable the auto refresh of coordination points variable in case registration keys are missing on any of CP servers.

Sample response file for configuring disk-based I/O fencing

Review the disk-based I/O fencing response file variables and their definitions.

See "Response file variables to configure disk-based I/O fencing" on page 125.

```
# Configuration Values:
#
our %CFG;
$CFG{fencing_config_cpagent}=1;
$CFG{fencing_auto_refresh_reg}=1;
$CFG{fencing_cpagent_monitor_freq}=5;
$CFG{fencing_cpagentgrp}="vxfen";
$CFG{fencing_dgname}="fencingdg1";
$CFG{fencing_newdg_disks}=[ qw(emc_clariion0_155
 emc_clariion0_162 emc_clariion0_163) ];
$CFG{fencing_option}=2;
$CFG{opt}{configure}=1;
$CFG{opt}{fencing}=1;

$CFG{activecomponent}="SFRAC742";
$CFG{systems}=[ qw(sys1sys2)];
$CFG{vcs_clusterid}=32283;
$CFG{vcs_clustername}="clus1";
1;
```

Response file variables to configure server-based I/O fencing

You can use a coordination point server-based fencing response file to configure server-based customized I/O fencing.

Table 6-2 lists the fields in the response file that are relevant for server-based customized I/O fencing.

Table 6-2 Coordination point server (CP server) based fencing response file definitions

Response file field	Definition
CFG {fencing_config_cpagent}	Enter '1' or '0' depending upon whether you want to configure the Coordination Point agent using the installer or not. Enter "0" if you do not want to configure the Coordination Point agent using the installer. Enter "1" if you want to use the installer to configure the Coordination Point agent.
CFG {fencing_cpagentgrp}	Name of the service group which will have the Coordination Point agent resource as part of it. **Note:** This field is obsolete if the `fencing_config_cpagent` field is given a value of '0'.
CFG {fencing_cps}	Virtual IP address or Virtual hostname of the CP servers.

Table 6-2 Coordination point server (CP server) based fencing response file definitions *(continued)*

Response file field	Definition
CFG {fencing_reusedg}	This response file field indicates whether to reuse an existing DG name for the fencing configuration in customized fencing (CP server and coordinator disks). Enter either a "1" or "0". Entering a "1" indicates reuse, and entering a "0" indicates do not reuse. When reusing an existing DG name for the mixed mode fencing configuration. you need to manually add a line of text , such as "$CFG{fencing_reusedg}=0" or "$CFG{fencing_reusedg}=1" before proceeding with a silent installation.
CFG {fencing_dgname}	The name of the disk group to be used in the customized fencing, where at least one disk is being used.
CFG {fencing_disks}	The disks being used as coordination points if any.
CFG {fencing_ncp}	Total number of coordination points being used, including both CP servers and disks.
CFG {fencing_ndisks}	The number of disks being used.
CFG {fencing_cps_vips}	The virtual IP addresses or the fully qualified host names of the CP server.
CFG {fencing_cps_ports}	The port that the virtual IP address or the fully qualified host name of the CP server listens on.

Table 6-2 Coordination point server (CP server) based fencing response file definitions *(continued)*

Response file field	Definition
CFG{fencing_option}	Specifies the I/O fencing configuration mode. ■ 1—Coordination Point Server-based I/O fencing ■ 2—Coordinator disk-based I/O fencing ■ 3—Disabled-based I/O fencing ■ 4—Online fencing migration ■ 5—Refresh keys/registrations on the existing coordination points ■ 6—Change the order of existing coordination points ■ 7—Majority-based fencing (Required)
CFG{fencing_auto_refresh_reg}	Enable this variable if registration keys are missing on any of the CP servers.

Sample response file for configuring server-based I/O fencing

The following is a sample response file used for server-based I/O fencing:

```
# Configuration Values:
# our %CFG;
$CFG{fencing_config_cpagent}=0;
$CFG{fencing_cps}=[ qw(10.200.117.145) ];
$CFG{fencing_cps_vips}{"10.200.117.145"}=[ qw(10.200.117.145) ];
$CFG{fencing_dgname}="vxfencoorddg";
$CFG{fencing_disks}=[ qw(emc_clariion0_37 emc_clariion0_13
emc_clariion0_12) ];
$CFG{fencing_scsi3_disk_policy}="dmp";
$CFG{fencing_ncp}=3;
$CFG{fencing_ndisks}=2;
$CFG{fencing_cps_ports}{"10.200.117.145"}=443;
$CFG{fencing_reusedg}=1;
$CFG{opt}{configure}=1;
```

```
$CFG{opt}{fencing}=1;
$CFG{prod}="AVAILABILITY742";
$CFG{systems}=[ qw(sys1 sys2) ];
$CFG{vcs_clusterid}=1256;
$CFG{vcs_clustername}="clus1";
$CFG{fencing_option}=1;
```

Response file variables to configure non-SCSI-3 I/O fencing

Table 6-3 lists the fields in the response file that are relevant for non-SCSI-3 I/O fencing.

Table 6-3 Non-SCSI-3 I/O fencing response file definitions

Response file field	Definition
CFG{non_scsi3_fencing}	Defines whether to configure non-SCSI-3 I/O fencing. Valid values are 1 or 0. Enter 1 to configure non-SCSI-3 I/O fencing.
CFG {fencing_config_cpagent}	Enter '1' or '0' depending upon whether you want to configure the Coordination Point agent using the installer or not. Enter "0" if you do not want to configure the Coordination Point agent using the installer. Enter "1" if you want to use the installer to configure the Coordination Point agent. **Note:** This variable does not apply to majority-based fencing.
CFG {fencing_cpagentgrp}	Name of the service group which will have the Coordination Point agent resource as part of it. **Note:** This field is obsolete if the `fencing_config_cpagent` field is given a value of '0'. This variable does not apply to majority-based fencing.
CFG {fencing_cps}	Virtual IP address or Virtual hostname of the CP servers. **Note:** This variable does not apply to majority-based fencing.

Table 6-3 Non-SCSI-3 I/O fencing response file definitions *(continued)*

Response file field	Definition
CFG {fencing_cps_vips}	The virtual IP addresses or the fully qualified host names of the CP server. **Note:** This variable does not apply to majority-based fencing.
CFG {fencing_ncp}	Total number of coordination points (CP servers only) being used. **Note:** This variable does not apply to majority-based fencing.
CFG {fencing_cps_ports}	The port of the CP server that is denoted by *cps* . **Note:** This variable does not apply to majority-based fencing.
CFG{fencing_auto_refresh_reg}	Enable this variable if registration keys are missing on any of the CP servers.

Sample response file for configuring non-SCSI-3 I/O fencing

The following is a sample response file used for non-SCSI-3 I/O fencing :

```
# Configuration Values:
# our %CFG;
$CFG{fencing_config_cpagent}=0;
$CFG{fencing_cps}=[ qw(10.198.89.251 10.198.89.252 10.198.89.253) ];
$CFG{fencing_cps_vips}{"10.198.89.251"}=[ qw(10.198.89.251) ];
$CFG{fencing_cps_vips}{"10.198.89.252"}=[ qw(10.198.89.252) ];
$CFG{fencing_cps_vips}{"10.198.89.253"}=[ qw(10.198.89.253) ];
$CFG{fencing_ncp}=3;
$CFG{fencing_ndisks}=0;
$CFG{fencing_cps_ports}{"10.198.89.251"}=443;
$CFG{fencing_cps_ports}{"10.198.89.252"}=443;
$CFG{fencing_cps_ports}{"10.198.89.253"}=443;
$CFG{non_scsi3_fencing}=1;
$CFG{opt}{configure}=1;
$CFG{opt}{fencing}=1;
$CFG{prod}="AVAILABILITY742";
```

```
$CFG{systems}=[ qw(sys1 sys2) ];
$CFG{vcs_clusterid}=1256;
$CFG{vcs_clustername}="clus1";
$CFG{fencing_option}=1;
```

Response file variables to configure majority-based I/O fencing

Table 6-4 lists the response file variables that specify the required information to configure disk-based I/O fencing for VCS.

Table 6-4 Response file variables specific to configuring majority-based I/O fencing

Variable	List or Scalar	Description
CFG{opt}{fencing}	Scalar	Performs the I/O fencing configuration. (Required)
CFG{fencing_option}	Scalar	Specifies the I/O fencing configuration mode. ■ 1—Coordination Point Server-based I/O fencing ■ 2—Coordinator disk-based I/O fencing ■ 3—Disabled-based fencing ■ 4—Online fencing migration ■ 5—Refresh keys/registrations on the existing coordination points ■ 6—Change the order of existing coordination points ■ 7—Majority-based fencing (Required)

Sample response file for configuring majority-based I/O fencing

```
# Configuration Values:
# our %CFG;
```

```
$CFG{fencing_option}=7;
$CFG{config_majority_based_fencing}=1;
$CFG{opt}{configure}=1;
$CFG{opt}{fencing}=1;
$CFG{prod}="AVAILABILITY742";
$CFG{systems}=[ qw(sys1  sys2) ];
$CFG{vcs_clusterid}=59082;
$CFG{vcs_clustername}="clus1";
```

Section 3

Manual configuration

Manually configuring VCS

This chapter includes the following topics:

About configuring VCS manually

This section describes the procedures to manually configure VCS.

Note: For manually configuring VCS in single node mode, you can skip steps about configuring LLT manually and configuring GAB manually.

Configuring LLT manually

VCS uses the Low Latency Transport (LLT) protocol for all cluster communications as a high-performance, low-latency replacement for the IP stack. LLT has two major functions.

It handles the following tasks:

- Traffic distribution
- Heartbeat traffic

To configure LLT over Ethernet, perform the following steps on each node in the cluster:

- Set up the file /etc/llthosts.
  See "Setting up /etc/llthosts for a manual installation" on page 138.
- Set up the file /etc/llttab.
  See "Setting up /etc/llttab for a manual installation" on page 139.
- Edit the following file on each node in the cluster to change the values of the LLT_START and the LLT_STOP environment variables to 1:
  /etc/sysconfig/llt

You can also configure LLT over UDP.

See "Using the UDP layer for LLT" on page 290.

You can also configure LLT over RDMA.

See "Using LLT over RDMA: supported use cases " on page 327.

Setting up /etc/llthosts for a manual installation

The file llthosts(4) is a database. It contains one entry per system that links the LLT system ID (in the first column) with the LLT host name. You must ensure that contents of this file are identical on all the nodes in the cluster. A mismatch of the contents of the file can cause indeterminate behavior in the cluster.

Use vi or another editor, to create the file /etc/llthosts that contains the entries that resemble:

```
0 sys1
1 sys2
```

Setting up /etc/llttab for a manual installation

The /etc/llttab file must specify the system's ID number (or its node name), its cluster ID, and the network links that correspond to the system. In addition, the file can contain other directives. Refer also to the sample llttab file in /opt/VRTSllt.

See "About LLT directives in /etc/llttab file" on page 139.

Use vi or another editor to create the file /etc/llttab that contains the entries that resemble:

```
set-node node_name
set-cluster cluster_id
link eth1 eth-MAC_address - ether - -
link eth2 eth-MAC_address - ether - -
```

The first line must identify the system where the file exists. In the example, the value for `set-node` can be: sys1 or 0. The next line, beginning with the `set-cluster` command, identifies the cluster number, which must be a unique number when more than one cluster is configured on the same physical network connection. The next two lines, beginning with the `link` command, identify the two private network cards that the LLT protocol uses. The order of directives must be the same as in the sample `llttab` file in `/opt/VRTSllt`.

If you use different media speed for the private NICs, Veritas recommends that you configure the NICs with lesser speed as low-priority links to enhance LLT performance. For example:

Use vi or another editor to create the file /etc/llttab that contains the entries that resemble:

```
set-node node name
set-cluster cluster_id
link eth1 eth-MAC_address - ether - -
link eth2 eth-MAC_address - ether - -
link-lowpri eth3 eth-MAC_address - ether - -
```

About LLT directives in /etc/llttab file

Table 7-1 lists the LLT directives in /etc/llttab file for LLT over Ethernet.

Table 7-1 LLT directives

Directive	Description
`set-node`	Assigns the system ID or symbolic name. The system ID number must be unique for each system in the cluster, and must be in the range 0-63. The symbolic name corresponds to the system ID, which is in /etc/llthosts file. Note that LLT fails to operate if any systems share the same ID.
`set-cluster`	Assigns a unique cluster number. Use this directive when more than one cluster is configured on the same physical network connection. LLT uses a default cluster number of zero.
`link`	Attaches LLT to a network interface. At least one link is required, and up to eight are supported. LLT distributes network traffic evenly across all available network connections unless you mark the link as low-priority using the link-lowpri directive or you configured LLT to use destination-based load balancing. The first argument to link is a user-defined tag shown in the `lltstat(1M)` output to identify the link. It may also be used in `llttab` to set optional static MAC addresses. The second argument to link specifies the network interface to use. For bonds or vlan interfaces, use the interface name. For standard network interfaces, Veritas recommends the usage of eth-*mac* to specify the corresponding network interface. The remaining four arguments to link are defaults; these arguments should be modified only in advanced configurations. There should be one link directive for each network interface. LLT uses an unregistered Ethernet SAP of 0xcafe. If the SAP is unacceptable, refer to the `llttab(4)` manual page for information on how to customize SAP. Note that IP addresses do not need to be assigned to the network device; LLT does not use IP addresses in LLT over Ethernet mode.
`link-lowpri`	Use this directive in place of `link` for public network interfaces. This directive prevents VCS communication on the public network until the network is the last link, and reduces the rate of heartbeat broadcasts. If you use private NICs with different speed, use "link-lowpri" directive in place of "link" for all links with lower speed. Use the "link" directive only for the private NIC with higher speed to enhance LLT performance. LLT uses low-priority network links for VCS communication only when other links fail.

For more information about the LLT directives, refer to the `llttab`(4) manual page.

Additional considerations for LLT for a manual installation

You must attach each network interface that is configured for LLT to a separate and distinct physical network.

Configuring GAB manually

VCS uses the Group Membership Services/Atomic Broadcast (GAB) protocol for cluster membership and reliable cluster communications. GAB has two major functions.

It handles the following tasks:

- Cluster membership
- Cluster communications

To configure GAB

1 Set up an /etc/gabtab configuration file on each node in the cluster using vi or another editor. The following example shows an /etc/gabtab file:

```
/sbin/gabconfig -c -nN
```

Where the `-c` option configures the driver for use. The `-n`N option specifies that the cluster is not formed until at least N systems are ready to form the cluster. Veritas recommends that you set N to be the total number of systems in the cluster.

Warning: Veritas does not recommend the use of the `-c -x` option or `-x` option for `/sbin/gabconfig`. Using `-c -x` or `-x` can lead to a split-brain condition.

2 Edit the following file on each node in the cluster to change the values of the GAB_START and the GAB_STOP environment variables to 1:

/etc/sysconfig/gab

Configuring VCS manually

VCS configuration requires the types.cf and main.cf files on each system in the cluster. Both of the files are in the /etc/VRTSvcs/conf/config directory.

main.cf file

The main.cf configuration file requires the following minimum essential elements:

- An "include" statement that specifies the file, types.cf, which defines the VCS bundled agent resource type definitions.
- The name of the cluster.
- The name of the systems that make up the cluster.

types.cf file

Note that the "include" statement in main.cf refers to the types.cf file. This text file describes the VCS bundled agent resource type definitions. During new installations, the types.cf file is automatically copied in to the /etc/VRTSvcs/conf/config directory.

When you manually install VCS, the file /etc/VRTSvcs/conf/config/main.cf contains only the line:

```
include "types.cf"
```

For a full description of the main.cf file, and how to edit and verify it, refer to the *Cluster Server Administrator's Guide*.

To configure VCS manually

1. Log on as superuser, and move to the directory that contains the configuration file:

   ```
   # cd /etc/VRTSvcs/conf/config
   ```

2. Use vi or another text editor to edit the main.cf file, defining your cluster name and system names. Refer to the following example.

   An example main.cf for a two-node cluster:

   ```
   include "types.cf"
   cluster VCSCluster2 ( )
   system sys1 ( )
   system sys2 ( )
   ```

   An example main.cf for a single-node cluster:

   ```
   include "types.cf"
   cluster VCSCluster1 ( )
   system sn1 ( )
   ```

3. Save and close the main.cf file.

Configuring the cluster UUID when creating a cluster manually

You need to configure the cluster UUID when you manually create a cluster.

To configure the cluster UUID when you create a cluster manually

- On one node in the cluster, perform the following command to populate the cluster UUID on each node in the cluster.

  ```
  # /opt/VRTSvcs/bin/uuidconfig.pl -clus -configure nodeA
        nodeB ... nodeN
  ```

  Where *nodeA*, *nodeB*, through *nodeN* are the names of the cluster nodes.

Configuring VCS in single node mode

In addition to the steps mentioned in the manual configuration section, complete the following steps to configure VCS in single node mode.

See “Configuring VCS manually” on page 141.

To configure VCS in single node mode

1. Edit the following file to change the value of the ONENODE environment variable to **yes**.

   ```
   /etc/sysconfig/vcs
   ```

2. If the single node is intended only to manage applications, you can disable LLT, GAB, I/O fencing kernel modules.

   Note: Disabling VCS kernel modules means that you cannot make the applications highly available across multiple nodes.

See “Disabling LLT, GAB, and I/O fencing on a single node cluster” on page 143.

See “Enabling LLT, GAB, and I/O fencing” on page 145.

Disabling LLT, GAB, and I/O fencing on a single node cluster

This section discusses how to disable kernel modules on a single node VCS cluster.

Typically, LLT, GAB, and I/O fencing kernel modules are loaded on a node when you install VCS. However, you can disable LLT, GAB, and I/O fencing modules if you do not require high availability for the applications. You can continue to manage applications on the single node and use the application restart capabilities of VCS.

If you later decide to extend the cluster to multiple nodes, you can enable these modules and make the applications highly available across multiple nodes.

Note: If VCS engine hangs on the single node cluster with GAB disabled, GAB cannot detect the hang state and cannot take action to restart VCS. For such a condition, you need to detect that VCS engine has hung and take corrective action. For more information, refer to the 'About GAB client process failure' section in the *Cluster Server Administrator's Guide*.

See "Disabling LLT, GAB, and I/O fencing" on page 144.

Disabling LLT, GAB, and I/O fencing

Complete the following procedures to disable the kernel modules.

To disable I/O fencing

1 Edit the following file to set the value of `VXFEN_START` and `VXFEN_STOP` to **0** .

```
/etc/sysconfig/vxfen
```

2 Stop the I/O fencing module.

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

```
# systemctl stop vxfen
```

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

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

To disable GAB

1 Edit the following file to set the value of `GAB_START` and `GAB_STOP` to **0**.

```
/etc/sysconfig/gab
```

2 Stop the GAB module.

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

```
# systemctl stop gab
```

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

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

To disable LLT

1 Edit the following file to set the value of `LLT_START` and `LLT_STOP` to **0**.

```
/etc/sysconfig/llt
```

2 Stop the LLT module.

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

```
# systemctl stop llt
```

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

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

Enabling LLT, GAB, and I/O fencing

Complete the following procedures to enable the kernel modules.

To enable LLT:

1 Edit the following file to set the value of `LLT_START` and `LLT_STOP` to **1**.

```
/etc/sysconfig/llt
```

2 Start the LLT module.

For the supported Linux distributions:

```
# systemctl start llt
```

To enable GAB:

1 Edit the following file to set the value of `GAB_START` and `GAB_STOP` to **1**.

```
/etc/sysconfig/gab
```

2 Start the GAB module.

For the supported Linux distributions:

```
# systemctl start gab
```

To enable I/O fencing:

1 Edit the following file to set the value of `VXFEN_START` and `VXFEN_STOP` to **1** .

```
/etc/sysconfig/vxfen
```

2 Start the I/O fencing module.

For the supported Linux distributions:

```
# systemctl start vxfen
```

Starting LLT, GAB, and VCS after manual configuration

After you have configured LLT, GAB, and VCS, use the following procedures to start LLT, GAB, and VCS.

To start LLT

1 On each node, run the following command to start LLT:

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

```
# systemctl start llt
```

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

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

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

```
Loading LLT Driver...
Starting LLT:
LLT: loading module...
Loaded   kernel_version on kernel kernel_version
LLT: configuring module
where, kernel_version is the kernel version
of the Linux operating system
```

2 On each node, run the following command to verify that LLT is running:

```
# /sbin/lltconfig
LLT is running
```

To start GAB

1. On each node, run the following command to start GAB:

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

   ```
   # systemctl start gab
   ```

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

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

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

   ```
   GAB: Starting
   GAB: Starting Done
   ```

2. On each node, run the following command to verify that GAB is running:

   ```
   # /sbin/gabconfig -a
   GAB Port Memberships
   ===================================
   Port a gen a36e0003 membership 01
   ```

To start VCS

◆ On each node, type:

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

```
# systemctl start vcs
```

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

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

If VCS is configured correctly on each node, the engine log file at `/var/VRTSvcs/log/engine_A.log` resembles:

```
VCS NOTICE V-16-1-10619 ‘HAD’ starting on: sys1
VCS NOTICE V-16-1-10620 Waiting for local cluster configuration
status
VCS NOTICE V-16-1-10625 Local cluster configuration valid
VCS NOTICE V-16-1-11034 Registering for cluster membership
VCS NOTICE V-16-1-11035 Waiting for cluster membership
GAB INFO V-15-1-20036 Port h gen   265f06 membership ;1
GAB INFO V-15-1-20038 Port h gen   265f06 k_jeopardy 0
GAB INFO V-15-1-20040 Port h gen   265f06    visible 0
VCS INFO V-16-1-10077 Received new cluster membership
VCS NOTICE V-16-1-10082 System (sys1) is in Regular Membership
```

```
- Membership: 0x2
VCS NOTICE V-16-1-10073 Building from local configuration
VCS NOTICE V-16-1-10066 Entering RUNNING state
GAB INFO V-15-1-20036 Port h gen   265f07 membership 01
VCS INFO V-16-1-10077 Received new cluster membership
VCS NOTICE V-16-1-10082 System (sys2) is in Regular Membership
- Membership: 0x3
```

About configuring cluster using VCS Cluster Configuration wizard

Consider the following before configuring a cluster using VCS Cluster Configuration wizard

- The VCS Cluster Configuration wizard allows you to configure a VCS cluster and add a node to the cluster.
  See "Configuring a cluster by using the VCS cluster configuration wizard" on page 150.
- Veritas recommends that you first configure application monitoring using the wizard before using VCS commands to add additional components or modify the existing configuration. Apart from configuring application availability, the wizard also sets up the other components required for successful application monitoring.
- In VMware environment, you must not suspend a system if an application is currently online on that machine. If you suspend a system, VCS moves the disks along with the application to another system. Later, when you try to restore the suspended system, VMware does not allow the operation because the disks that were attached before the system was suspended are no longer with the system.

Before configuring a VCS cluster using the VCS Cluster Configuration wizard

Ensure that you complete the following tasks before launching the VCS Cluster Configuration wizard to configure a VCS cluster:

- Install InfoScale Availability or InfoScale Enterprise on the system on which you want to configure the VCS cluster.
- You must have the following user privileges when you attempt to configure the VCS cluster:

  - Configure Application Monitoring (Admin) privileges when you launch the wizard from the vSphere client.
  - Admin role privileges if you launch the wizard through VOM
- Install the application and the associated components that you want to monitor on the system.
- If you have configured a firewall, ensure that your firewall settings allow access to ports used by Cluster Server installer, wizards, and services.
  Verify that the following ports are not blocked by the firewall:

Physical environment	5634, 14161, 14162, 14163, and 14164 At least one port from 14161, 14162, 14163, and 14164 must be open.

- You must not select bonded interfaces for cluster communication. A bonded interface is a logical NIC, formed by grouping several physical NICs together. All NICs in a bond have an identical MAC address, due to which you may experience the following issues:
  - Single Sign On (SSO) configuration failure.
  - The wizard may fail to discover the specified network adapters.
  - The wizard may fail to discover or validate the specified system name.
- In VMware environment, verify that the disks used by the application that you want to monitor are attached to non-shared controllers so that they can be detached from the system and attached to another system.
- The host name of the system must be resolvable through the DNS server or locally, using /etc/hosts file entries.

Launching the VCS Cluster Configuration wizard

You must launch the VCS Cluster Configuration wizard from the system where the disk residing on the shared datastore is attached.

You can launch the VCS Cluster Configuration wizard from:

- A browser window
  See Launching the VCS Cluster Configuration wizard from a browser window.

Launching the VCS Cluster Configuration wizard from a browser window

You can launch the VCS Cluster Configuration wizard from the Veritas High Availability view.

1 Open a browser window and enter the following URL:

   https://<IP_or_HostName>:5634/vcs/admin/application_health.html

   where <IP_or_HostName> is the IP address or host name of the system on which you want to configure the cluster.

2 Click the **Configure cluster** link on the Veritas High Availability view page to launch the wizard.

Note: At various stages of cluster configuration, the Veritas High Availability view offers different configuration options. These options launch appropriate wizard panels based on the tasks that you choose to perform.

See “Configuring a cluster by using the VCS cluster configuration wizard” on page 150.

See “Adding a system to a VCS cluster” on page 154.

Refer to the *Administering application monitoring from the Veritas High Availability view* section in *Cluster Server Administrator's Guide* for more information on the configurations possible from the Veritas High Availability view.

Configuring a cluster by using the VCS cluster configuration wizard

Perform the following steps to configure a Cluster Server (VCS) cluster by using the VCS Cluster Configuration wizard.

To configure a VCS cluster

1 Access the Veritas High Availability view (for any system belonging the required cluster).

   See “Launching the VCS Cluster Configuration wizard” on page 149.

2 Review the information on the Welcome panel and click **Next**.

   The Configuration Inputs panel appears.

   The local system is by default selected as a cluster system.

3 If you do not want to add more systems to the cluster, skip this step. You can add systems later using the same wizard.

To add a system to the cluster, click **Add System**.

In the Add System dialog box, specify the following details for the system that you want to add to the VCS cluster and click **OK**.

System Name or IP address	Specify the name or IP address of the system that you want to add to the VCS cluster.
User name	Specify the user account for the system. Typically, this is the root user. The root user should have the necessary privileges.
Password	Specify the password for the user account you specified.
Use the specified user account on all systems	Select this check box to use the specified user account on all the cluster systems that have the same user name and password.

4 On the Configuration Inputs panel, do one of the following actions:

- To add another system to the cluster, click Add System and repeat step 3.
- To modify the specified User name or Password for a cluster system, use the edit icon.
- Click **Next**

5 If you do not want to modify the security settings for the cluster, click **Next**, and proceed to step 7.

By default, the wizard configures single sign-on for secure cluster communication. If you want to modify the security settings for the cluster, click **Advanced Settings**.

6 In the Advanced settings dialog box, specify the following details and click **OK**.

Use Single Sign-on	Select to configure single sign-on using VCS Authentication Service for cluster communication. This option is enabled by default.
Use VCS user privileges	Select to configure a user with administrative privileges to the cluster. Specify the username and password and click OK.

7 On the Network Details panel, select the type of network protocol to configure the VCS cluster network links (Low Latency Transport or LLT module), and then specify the adapters for network communication.

The wizard configures the VCS cluster communication links using these adapters. You must select a minimum of two adapters per cluster system.

Note: By default, the LLT links are configured over Ethernet.

Select **Use MAC address for cluster communication (LLT over Ethernet)** or select **Use IP address for cluster communication (LLT over UDP)**, and specify the following details for each cluster system.

- To configure LLT over Ethernet, select the adapter for each network communication link. You must select a different network adapter for each communication link.
- To configure LLT over UDP, select the type of IP protocol (IPv4 or IPv6), and then specify the required details for each communication link.

Network Adapter	Select a network adapter for the communication links. You must select a different network adapter for each communication link.
IP Address	Displays the IP address.

Port	Specify a unique port number for each link. For IPv4 and IPv6, the port range is from 49152 to 65535. A specified port for a link is used for all the cluster systems on that link.
Subnet mask (IPv4)	Displays the subnet mask details.
Prefix (IPv6)	Displays the prefix details.

By default, one of the links is configured as a low-priority link on a public network interface. The second link is configured as a high-priority link.

To change a high-priority link to a low-priority link, click **Modify**. In the Modify low-priority link dialog box, select the link and click **OK**.

Note: Veritas recommends that you configure one of the links on a public network interface. You can assign the link on the public network interface as a low-priority link for minimal VCS cluster communication over the link.

8 On the Configuration Summary panel, specify a cluster name and unique cluster ID and then click **Validate**.

Note: If multiple clusters exist in your network, the wizard validates if the specified cluster ID is a unique cluster ID among all clusters accessible from the current system. Among clusters that are not accessible from the current system, you must ensure that the cluster ID you specified is unique

9 Review the VCS Cluster Configuration Details and then click **Next** to proceed with the configuration

10 On the Implementation panel, the wizard creates the VCS cluster.

The wizard displays the status of the configuration task. After the configuration is complete, click **Next**.

If the configuration task fails, click **Diagnostic information** to check the details of the failure. Rectify the cause of the failure and run the wizard again to configure the VCS cluster.

11 On the Finish panel, click **Finish** to complete the wizard workflow.

This completes the VCS cluster configuration.

Adding a system to a VCS cluster

Perform the following steps to add a system to a Cluster Server (VCS) cluster by using the VCS Cluster Configuration wizard.

The system from where you launch the wizard must be part of the cluster to which you want to add a new system.

To add a system to a VCS cluster

1 Access the Veritas High Availability view (for any system belonging to the required cluster).

See "Launching the VCS Cluster Configuration wizard" on page 149.

2 Click **Actions** > **Add System to VCS Cluster**.

The VCS Cluster Configuration Wizard is launched.

3 Review the information on the Welcome panel and click **Next**.

The Configuration Inputs panel appears, along with the cluster name, and a table of existing cluster systems.

4 To add a system to the cluster, click **Add System**.

5 In the Add System dialog box, specify the following details for the system that you want to add to the VCS cluster and click **OK**.

System Name or IP address	Specify the name or IP address of the system that you want to add to the VCS cluster.
User name	Specify the user account for the system. Typically, this is the root user. The root user should have the necessary privileges.
Password	Specify the password for the user account you specified.
Use the specified user account on all systems	Select this check box to use the specified user account on all the cluster systems that have the same user name and password.

6 On the Configuration Inputs panel, do one of the following actions:

- To add another system to the cluster, click **Add System** and repeat step 4.
- To modify the User name or Password for a cluster system, use the edit icon.

- Click **Next**

7 On the Network Details panel, specify the adapters for network communication (Low Latency Transport or LLT module of VCS) for the system. The wizard configures the VCS cluster communication links using these adapters. You must select a minimum of two adapters.

Note: You cannot modify the existing type of cluster communication (LLT over Ethernet or LLT over UDP).

- If the existing cluster uses LLT over Ethernet, select the adapter for each network communication link. You must select a different network adapter for each communication link.
- If the existing cluster uses LLT over UDP, select the type of IP protocol (IPv4 or IPv6), and then specify the required details for each communication link.

Network Adapter	Select a network adapter for the communication links. You must select a different network adapter for each communication link.
IP Address	Displays the IP address.
Port	Specify a unique port number for each link. For IPv4 and IPv6, the port range is from 49152 to 65535. A specified port for a link is used for all the cluster systems on that link.
Subnet mask (IPv4)	Displays the subnet mask details.
Prefix (IPv6)	Displays the prefix details.

By default, one of the links is configured as a low-priority link on a public network interface. The other link is configured as a high-priority link.

To change a high-priority link to a low-priority link, click **Modify**. In the Modify low-priority link dialog box, select the link and click **OK**.

Note: Veritas recommends that you configure one of the links on a public network interface. You can assign the link on the public network interface as a low-priority link for minimal VCS cluster communication over the link.

8. On the Configuration Summary panel, review the VCS Cluster Configuration Details.

9. On the Implementation panel, the wizard creates the VCS cluster.

   The wizard displays the status of the configuration task. After the configuration is complete, click **Next**.

   If the configuration task fails, click **Diagnostic information** to check the details of the failure. Rectify the cause of the failure and run the wizard again to add the required system to the VCS cluster.

10. On the Finish panel, click **Finish** to complete the wizard workflow.

Modifying the VCS configuration

After the successful installation of VCS, you can modify the configuration of VCS using several methods. You can dynamically modify the configuration from the command line, Veritas InfoScale Operations Manager, or the Cluster Manager (Java Console). For information on management tools, refer to the *Cluster Server Administrator's Guide*.

You can also edit the main.cf file directly. For information on the structure of the main.cf file, refer to the *Cluster Server Administrator's Guide*.

Configuring the ClusterService group

When you have installed VCS, and verified that LLT, GAB, and VCS work, you can create a service group to include the optional features. These features include the VCS notification components and the Global Cluster option. If you manually added VCS to your cluster systems, you must manually create the ClusterService group. You can refer to the configuration examples of a system with a ClusterService group. See the *Cluster Server Administrator's Guide* for more information.

See “Sample main.cf file for VCS clusters” on page 274.

Manually configuring the clusters for data integrity

This chapter includes the following topics:

- Setting up disk-based I/O fencing manually
- Setting up server-based I/O fencing manually
- Setting up non-SCSI-3 fencing in virtual environments manually
- Setting up majority-based I/O fencing manually

Setting up disk-based I/O fencing manually

Table 8-1 lists the tasks that are involved in setting up I/O fencing.

Table 8-1

Task	Reference
Initializing disks as VxVM disks	See "Initializing disks as VxVM disks" on page 79.
Identifying disks to use as coordinator disks	See "Identifying disks to use as coordinator disks" on page 158.
Checking shared disks for I/O fencing	See "Checking shared disks for I/O fencing" on page 84.
Setting up coordinator disk groups	See "Setting up coordinator disk groups" on page 158.
Creating I/O fencing configuration files	See "Creating I/O fencing configuration files" on page 159.

Table 8-1 *(continued)*

Task	Reference
Modifying VCS configuration to use I/O fencing	See "Modifying VCS configuration to use I/O fencing" on page 160.
Configuring CoordPoint agent to monitor coordination points	See "Configuring CoordPoint agent to monitor coordination points" on page 173.
Verifying I/O fencing configuration	See "Verifying I/O fencing configuration" on page 161.

Identifying disks to use as coordinator disks

Make sure you initialized disks as VxVM disks.

See "Initializing disks as VxVM disks" on page 79.

Review the following procedure to identify disks to use as coordinator disks.

To identify the coordinator disks

1 List the disks on each node.

For example, execute the following commands to list the disks:

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

2 Pick three SCSI-3 PR compliant shared disks as coordinator disks.

See "Checking shared disks for I/O fencing" on page 84.

Setting up coordinator disk groups

From one node, create a disk group named vxfencoorddg. This group must contain three disks or LUNs. You must also set the coordinator attribute for the coordinator disk group. VxVM uses this attribute to prevent the reassignment of coordinator disks to other disk groups.

Note that if you create a coordinator disk group as a regular disk group, you can turn on the coordinator attribute in Volume Manager.

Refer to the *Storage Foundation Administrator's Guide* for details on how to create disk groups.

The following example procedure assumes that the disks have the device names sdx, sdy, and sdz.

To create the vxfencoorddg disk group

1. On any node, create the disk group by specifying the device names:

   ```
   # vxdg init vxfencoorddg sdx sdy sdz
   ```

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

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

3. Deport the coordinator disk group:

   ```
   # vxdg deport vxfencoorddg
   ```

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

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

5. Deport the disk group. Deporting the disk group prevents the coordinator disks from serving other purposes:

   ```
   # vxdg deport vxfencoorddg
   ```

Creating I/O fencing configuration files

After you set up the coordinator disk group, you must do the following to configure I/O fencing:

- Create the I/O fencing configuration file /etc/vxfendg
- Update the I/O fencing configuration file /etc/vxfenmode

To update the I/O fencing files and start I/O fencing

1. On each nodes, type:

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

   Do not use spaces between the quotes in the "vxfencoorddg" text.

   This command creates the /etc/vxfendg file, which includes the name of the coordinator disk group.

2. On all cluster nodes specify the use of DMP disk policy in the `/etc/vxfenmode` file.

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

3 To check the updated /etc/vxfenmode configuration, enter the following command on one of the nodes. For example:

```
# more /etc/vxfenmode
```

4 Ensure that you edit the following file on each node in the cluster to change the values of the VXFEN_START and the VXFEN_STOP environment variables to 1:

/etc/sysconfig/vxfen

Modifying VCS configuration to use I/O fencing

After you add coordination points and configure I/O fencing, add the UseFence = SCSI3 cluster attribute to the VCS configuration file /etc/VRTSvcs/conf/config/main.cf.

If you reset this attribute to UseFence = None, VCS does not make use of I/O fencing abilities while failing over service groups. However, I/O fencing needs to be disabled separately.

To modify VCS configuration to enable I/O fencing

1 Save the existing configuration:

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

2 Stop VCS on all nodes:

```
# hastop -all
```

3 To ensure High Availability has stopped cleanly, run:

```
gabconfig -a
```

In the output of the commands, check that Port h is not present.

4 If the I/O fencing driver vxfen is already running, stop the I/O fencing driver.

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

```
# systemctl stop vxfen
```

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

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

5 Make a backup of the main.cf file on all the nodes:

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

6. On one node, use vi or another text editor to edit the main.cf file. To modify the list of cluster attributes, add the UseFence attribute and assign its value as SCSI3.

   ```
   cluster clus1(
   UserNames = { admin = "cDRpdxPmHpzS." }
   Administrators = { admin }
   HacliUserLevel = COMMANDROOT
   CounterInterval = 5
   UseFence = SCSI3
   )
   ```

   Regardless of whether the fencing configuration is disk-based or server-based, the value of the cluster-level attribute UseFence is set to SCSI3.

7. Save and close the file.

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

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

9. Start the I/O fencing driver and VCS. Perform the following steps on each node:

   - Start the I/O fencing driver.
     The vxfen startup script also invokes the `vxfenconfig` command, which configures the vxfen driver to start and use the coordination points that are listed in /etc/vxfentab.
     For RHEL 7, SLES 12, and supported RHEL distributions:

     ```
     # systemctl start vxfen
     ```

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

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

   - Start VCS on the node where main.cf is modified.

     ```
     # /opt/VRTS/bin/hastart
     ```

   - Start VCS on all other nodes once VCS on first node reaches RUNNING state.

     ```
     # /opt/VRTS/bin/hastart
     ```

Verifying I/O fencing configuration

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

To verify I/O fencing configuration

1 On one of the nodes, type:

```
# vxfenadm -d
```

Output similar to the following appears if the fencing mode is SCSI3 and the SCSI3 disk policy is dmp:

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

Fencing Protocol Version: 201
Fencing Mode: SCSI3
Fencing SCSI3 Disk Policy: dmp
Cluster Members:

   * 0 (sys1)
   1 (sys2)

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

2 Verify that the disk-based I/O fencing is using the specified disks.

```
# vxfenconfig -l
```

Setting up server-based I/O fencing manually

Tasks that are involved in setting up server-based I/O fencing manually include:

Table 8-2 Tasks to set up server-based I/O fencing manually

Task	Reference
Preparing the CP servers for use by the VCS cluster	See "Preparing the CP servers manually for use by the VCS cluster" on page 163.
Generating the client key and certificates on the client nodes manually	See "Generating the client key and certificates manually on the client nodes " on page 165.

Table 8-2 Tasks to set up server-based I/O fencing manually *(continued)*

Task	Reference
Modifying I/O fencing configuration files to configure server-based I/O fencing	See "Configuring server-based fencing on the VCS cluster manually" on page 167.
Modifying VCS configuration to use I/O fencing	See "Modifying VCS configuration to use I/O fencing" on page 160.
Configuring Coordination Point agent to monitor coordination points	See "Configuring CoordPoint agent to monitor coordination points" on page 173.
Verifying the server-based I/O fencing configuration	See "Verifying server-based I/O fencing configuration" on page 175.

Preparing the CP servers manually for use by the VCS cluster

Use this procedure to manually prepare the CP server for use by the VCS cluster or clusters.

Table 8-3 displays the sample values used in this procedure.

Table 8-3 Sample values in procedure

CP server configuration component	Sample name
CP server	cps1
Node #1 - VCS cluster	sys1
Node #2 - VCS cluster	sys2
Cluster name	clus1
Cluster UUID	{f0735332-1dd1-11b2}

To manually configure CP servers for use by the VCS cluster

1. Determine the cluster name and uuid on the VCS cluster.

   For example, issue the following commands on one of the VCS cluster nodes (sys1):

   ```
   # grep cluster /etc/VRTSvcs/conf/config/main.cf
   ```

   ```
   cluster clus1
   ```

   ```
   # cat /etc/vx/.uuids/clusuuid
   ```

   ```
   {f0735332-1dd1-11b2-bb31-00306eea460a}
   ```

2. Use the `cpsadm` command to check whether the VCS cluster and nodes are present in the CP server.

   For example:

```
# cpsadm -s cps1.example.com -a list_nodes

ClusName  UUID                                Hostname(Node ID) Registered
clus1  {f0735332-1dd1-11b2-bb31-00306eea460a} sys1(0)         0
clus1  {f0735332-1dd1-11b2-bb31-00306eea460a} sys2(1)         0
```

   If the output does not show the cluster and nodes, then add them as described in the next step.

   For detailed information about the `cpsadm` command, see the *Cluster Server Administrator's Guide.*

3 Add the VCS cluster and nodes to each CP server.

For example, issue the following command on the CP server (cps1.example.com) to add the cluster:

```
# cpsadm -s cps1.example.com -a add_clus\
 -c clus1  -u {f0735332-1dd1-11b2}
```

```
Cluster clus1 added successfully
```

Issue the following command on the CP server (cps1.example.com) to add the first node:

```
# cpsadm -s cps1.example.com -a add_node\
 -c clus1 -u {f0735332-1dd1-11b2} -h sys1 -n0
```

```
Node 0 (sys1) successfully added
```

Issue the following command on the CP server (cps1.example.com) to add the second node:

```
# cpsadm -s cps1.example.com -a add_node\
 -c clus1 -u {f0735332-1dd1-11b2} -h sys2 -n1
```

```
Node 1 (sys2) successfully added
```

See “Generating the client key and certificates manually on the client nodes ” on page 165.

Generating the client key and certificates manually on the client nodes

The client node that wants to connect to a CP server using HTTPS must have a private key and certificates signed by the Certificate Authority (CA) on the CP server

The client uses its private key and certificates to establish connection with the CP server. The key and the certificate must be present on the node at a predefined location. Each client has one client certificate and one CA certificate for every CP server, so, the certificate files must follow a specific naming convention. Distinct certificate names help the `cpsadm` command to identify which certificates have to be used when a client node connects to a specific CP server.

The certificate names must be as follows: ca_*cps-vip*.crt and client _*cps-vip*.crt

Where, *cps-vip* is the VIP or FQHN of the CP server listed in the `/etc/vxfenmode` file. For example, for a sample VIP, *192.168.1.201*, the corresponding certificate name is ca_*192.168.1.201*.

To manually set up certificates on the client node

1 Create the directory to store certificates.

```
# mkdir -p /var/VRTSvxfen/security/keys
/var/VRTSvxfen/security/certs
```

Note: Since the openssl utility might not be available on client nodes, Veritas recommends that you access the CP server using SSH to generate the client keys or certificates on the CP server and copy the certificates to each of the nodes.

2 Generate the private key for the client node.

```
# /opt/VRTSperl/non-perl-libs/bin/openssl genrsa -out
client_private.key 2048
```

3 Generate the client CSR for the cluster. CN is the UUID of the client's cluster.

```
# /opt/VRTSperl/non-perl-libs/bin/openssl req -new -key -sha256
client_private.key\

-subj '/C=countryname/L=localityname/OU=COMPANY/CN=CLUS_UUID'\

-out client_192.168.1.201.csr
```

Where, *countryname* is the country code, *localityname* is the city, *COMPANY* is the name of the company, and *CLUS_UUID* is the certificate name.

4 Generate the client certificate by using the CA key and the CA certificate. Run this command from the CP server.

```
# /opt/VRTSperl/non-perl-libs/bin/openssl x509 -req -days days
-sha256 -in client_192.168.1.201.csr\

-CA /var/VRTScps/security/certs/ca.crt -CAkey\

/var/VRTScps/security/keys/ca.key -set_serial 01 -out
client_192.168.10.1.crt
```

Where, *days* is the days you want the certificate to remain valid, *192.168.1.201* is the VIP or FQHN of the CP server.

5 Copy the client key, client certificate, and CA certificate to each of the client nodes at the following location.

Copy the client key at `/var/VRTSvxfen/security/keys/client_private.key`. The client is common for all the client nodes and hence you need to generate it only once.

Copy the client certificate at `/var/VRTSvxfen/security/certs/client_192.168.1.201.crt`.

Copy the CA certificate at `/var/VRTSvxfen/security/certs/ca_192.168.1.201.crt`

Note: Copy the certificates and the key to all the nodes at the locations that are listed in this step.

6 If the client nodes need to access the CP server using the FQHN and or the host name, make a copy of the certificates you generated and replace the VIP with the FQHN or host name. Make sure that you copy these certificates to all the nodes.

7 Repeat the procedure for every CP server.

8 After you copy the key and certificates to each client node, delete the client keys and client certificates on the CP server.

Configuring server-based fencing on the VCS cluster manually

The configuration process for the client or VCS cluster to use CP server as a coordination point requires editing the `/etc/vxfenmode` file.

You need to edit this file to specify the following information for your configuration:

- Fencing mode
- Fencing mechanism
- Fencing disk policy (if applicable to your I/O fencing configuration)
- CP server or CP servers
- Coordinator disk group (if applicable to your I/O fencing configuration)
- Set the order of coordination points

Note: Whenever coordinator disks are used as coordination points in your I/O fencing configuration, you must create a disk group (vxfencoorddg). You must specify this disk group in the `/etc/vxfenmode` file.

See "Setting up coordinator disk groups" on page 158.

The customized fencing framework also generates the `/etc/vxfentab` file which has coordination points (all the CP servers and disks from disk group specified in `/etc/vxfenmode` file).

To configure server-based fencing on the VCS cluster manually

1. Use a text editor to edit the following file on each node in the cluster:

   ```
   /etc/sysconfig/vxfen
   ```

   You must change the values of the `VXFEN_START` and the `VXFEN_STOP` environment variables to 1.

2. Use a text editor to edit the `/etc/vxfenmode` file values to meet your configuration specifications.

   - If your server-based fencing configuration uses a single highly available CP server as its only coordination point, make sure to add the `single_cp=1` entry in the `/etc/vxfenmode` file.
   - If you want the `vxfen` module to use a specific order of coordination points during a network partition scenario, set the `vxfen_honor_cp_order` value to be **1**. By default, the parameter is disabled.

   The following sample file output displays what the `/etc/vxfenmode` file contains:

   See "Sample vxfenmode file output for server-based fencing" on page 168.

3. After editing the `/etc/vxfenmode` file, run the vxfen init script to start fencing.

   For example:

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

   ```
   # systemctl start vxfen
   ```

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

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

Sample vxfenmode file output for server-based fencing

The following is a sample vxfenmode file for server-based fencing:

```
#
```

```
# vxfen_mode determines in what mode VCS I/O Fencing should work.
#
# available options:
# scsi3      - use scsi3 persistent reservation disks
# customized - use script based customized fencing
# disabled   - run the driver but don't do any actual fencing
#
vxfen_mode=customized

# vxfen_mechanism determines the mechanism for customized I/O
# fencing that should be used.
#
# available options:
# cps      - use a coordination point server with optional script
#            controlled scsi3 disks
#
vxfen_mechanism=cps

#
# scsi3_disk_policy determines the way in which I/O fencing
# communicates with the coordination disks. This field is
# required only if customized coordinator disks are being used.
#
# available options:
# dmp - use dynamic multipathing
#
scsi3_disk_policy=dmp

#
# vxfen_honor_cp_order determines the order in which vxfen
# should use the coordination points specified in this file.
#
# available options:
# 0 - vxfen uses a sorted list of coordination points specified
# in this file,
# the order in which coordination points are specified does not matter.
#   (default)
# 1 - vxfen uses the coordination points in the same order they are
#    specified in this file

# Specify 3 or more odd number of coordination points in this file,
# each one in its own line. They can be all-CP servers,
# all-SCSI-3 compliant coordinator disks, or a combination of
# CP servers and SCSI-3 compliant coordinator disks.
# Please ensure that the CP server coordination points
# are numbered sequentially and in the same order
# on all the cluster nodes.
#
# Coordination Point Server(CPS) is specified as follows:
#
#  cps<number>=[<vip/vhn>]:<port>
#
# If a CPS supports multiple virtual IPs or virtual hostnames
# over different subnets, all of the IPs/names can be specified
# in a comma separated list as follows:
#
# cps<number>=[<vip_1/vhn_1>]:<port_1>,[<vip_2/vhn_2>]:<port_2>,
...,[<vip_n/vhn_n>]:<port_n>
#
# Where,
# <number>
#  is the serial number of the CPS as a coordination point; must
#  start with 1.
# <vip>
#  is the virtual IP address of the CPS, must be specified in
#  square brackets ("[]").
# <vhn>
#  is the virtual hostname of the CPS, must be specified in square
#  brackets ("[]").
# <port>
#  is the port number bound to a particular <vip/vhn> of the CPS.
#  It is optional to specify a <port>. However, if specified, it
#  must follow a colon (":") after <vip/vhn>. If not specified, the
#  colon (":") must not exist after <vip/vhn>.
#
# For all the <vip/vhn>s which do not have a specified <port>,
# a default port can be specified as follows:
#
# port=<default_port>
#
#  Where <default_port> is applicable to all the <vip/vhn>s for
#  which a <port> is not specified. In other words, specifying
#  <port> with a <vip/vhn> overrides the <default_port> for that
#  <vip/vhn>. If the <default_port> is not specified, and there
#  are <vip/vhn>s for which <port> is not specified, then port
#  number 14250 will be used for such <vip/vhn>s.
```

```
#
# Example of specifying CP Servers to be used as coordination points:
# port=57777
# cps1=[192.168.0.23],[192.168.0.24]:58888,[cps1.company.com]
#  cps2=[192.168.0.25]
# cps3=[cps2.company.com]:59999
#
# In the above example,
# - port 58888 will be used for vip [192.168.0.24]
# - port 59999 will be used for vhn [cps2.company.com], and
# - default port 57777 will be used for all remaining <vip/vhn>s:
#    [192.168.0.23]
#    [cps1.company.com]
#    [192.168.0.25]
# - if default port 57777 were not specified, port 14250
# would be used for all remaining <vip/vhn>s:
#    [192.168.0.23]
#    [cps1.company.com]
#    [192.168.0.25]
#
# SCSI-3 compliant coordinator disks are specified as:
#
#  vxfendg=<coordinator disk group name>
# Example:
#  vxfendg=vxfencoorddg
#
# Examples of different configurations:
#  1. All CP server coordination points
# cps1=
# cps2=
# cps3=
#
# 2. A combination of CP server and a disk group having two SCSI-3
# coordinator disks
# cps1=
# vxfendg=
# Note: The disk group specified in this case should have two disks
#
# 3. All SCSI-3 coordinator disks
# vxfendg=
# Note: The disk group specified in case should have three disks
# cps1=[cps1.company.com]
# cps2=[cps2.company.com]
# cps3=[cps3.company.com]
# port=443
```

Table 8-4 defines the vxfenmode parameters that must be edited.

Table 8-4 vxfenmode file parameters

vxfenmode File Parameter	Description
vxfen_mode	Fencing mode of operation. This parameter must be set to “customized”.
vxfen_mechanism	Fencing mechanism. This parameter defines the mechanism that is used for fencing. If one of the three coordination points is a CP server, then this parameter must be set to “cps”.
scsi3_disk_policy	Configure the vxfen module to use DMP devices, "dmp". **Note:** The configured disk policy is applied on all the nodes.

cps1, cps2, or vxfendg	Coordination point parameters.
	Enter either the virtual IP address or the FQHN (whichever is accessible) of the CP server.
	`cps<number>=[virtual_ip_address/virtual_host_name]:port`
	Where *port* is optional. The default port value is 443.
	If you have configured multiple virtual IP addresses or host names over different subnets, you can specify these as comma-separated values. For example:
	`cps1=[192.168.0.23],[192.168.0.24]:58888, [cps1.company.com]`
	Note: Whenever coordinator disks are used in an I/O fencing configuration, a disk group has to be created (vxfencoorddg) and specified in the /etc/vxfenmode file. Additionally, the customized fencing framework also generates the /etc/vxfentab file which specifies the security setting and the coordination points (all the CP servers and the disks from disk group specified in /etc/vxfenmode file).

Table 8-4 vxfenmode file parameters *(continued)*

vxfenmode File Parameter	Description
port	Default port for the CP server to listen on. If you have not specified port numbers for individual virtual IP addresses or host names, the default port number value that the CP server uses for those individual virtual IP addresses or host names is 443. You can change this default port value using the port parameter.
single_cp	Value 1 for single_cp parameter indicates that the server-based fencing uses a single highly available CP server as its only coordination point. Value 0 for single_cp parameter indicates that the server-based fencing uses at least three coordination points.
vxfen_honor_cp_order	Set the value to 1 for vxfen module to use a specific order of coordination points during a network partition scenario. By default the parameter is disabled. The default value is 0.

Configuring CoordPoint agent to monitor coordination points

The following procedure describes how to manually configure the CoordPoint agent to monitor coordination points.

The CoordPoint agent can monitor CP servers and SCSI-3 disks.

See the *Cluster Server Bundled Agents Reference Guide* for more information on the agent.

To configure CoordPoint agent to monitor coordination points

1. Ensure that your VCS cluster has been properly installed and configured with fencing enabled.

2. Create a parallel service group vxfen and add a coordpoint resource to the vxfen service group using the following commands:

   ```
   # haconf -makerw
   # hagrp -add vxfen
   # hagrp -modify vxfen SystemList sys1 0 sys2 1
   # hagrp -modify vxfen AutoFailOver 0
   # hagrp -modify vxfen Parallel 1
   # hagrp -modify vxfen SourceFile "./main.cf"
   # hares -add coordpoint CoordPoint vxfen
   # hares -modify coordpoint FaultTolerance 0
   # hares -override coordpoint LevelTwoMonitorFreq
   # hares -modify coordpoint LevelTwoMonitorFreq 5
   # hares -modify coordpoint Enabled 1
   # haconf -dump -makero
   ```

3. Configure the Phantom resource for the vxfen disk group.

   ```
   # haconf -makerw
   # hares -add RES_phantom_vxfen Phantom vxfen
   # hares -modify RES_phantom_vxfen Enabled 1
   # haconf -dump -makero
   ```

4 Verify the status of the agent on the VCS cluster using the `hares` commands. For example:

```
# hares -state coordpoint
```

The following is an example of the command and output::

```
# hares -state coordpoint
```

```
# Resource    Attribute   System   Value
coordpoint   State        sys1   ONLINE
coordpoint   State        sys2   ONLINE
```

5 Access the engine log to view the agent log. The agent log is written to the engine log.

The agent log contains detailed CoordPoint agent monitoring information; including information about whether the CoordPoint agent is able to access all the coordination points, information to check on which coordination points the CoordPoint agent is reporting missing keys, etc.

To view the debug logs in the engine log, change the dbg level for that node using the following commands:

```
# haconf -makerw
```

```
# hatype -modify Coordpoint LogDbg 10
```

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

The agent log can now be viewed at the following location:

/var/VRTSvcs/log/engine_A.log

Note: The Coordpoint agent is always in the online state when the I/O fencing is configured in the majority or the disabled mode. For both these modes the I/O fencing does not have any coordination points to monitor. Thereby, the Coordpoint agent is always in the online state.

Verifying server-based I/O fencing configuration

Follow the procedure described below to verify your server-based I/O fencing configuration.

To verify the server-based I/O fencing configuration

1 Verify that the I/O fencing configuration was successful by running the `vxfenadm` command. For example, run the following command:

```
# vxfenadm -d
```

Note: For troubleshooting any server-based I/O fencing configuration issues, refer to the *Cluster Server Administrator's Guide*.

2 Verify that I/O fencing is using the specified coordination points by running the `vxfenconfig` command. For example, run the following command:

```
# vxfenconfig -l
```

If the output displays `single_cp=1`, it indicates that the application cluster uses a CP server as the single coordination point for server-based fencing.

Setting up non-SCSI-3 fencing in virtual environments manually

To manually set up I/O fencing in a non-SCSI-3 PR compliant setup

1 Configure I/O fencing either in majority-based fencing mode with no coordination points or in server-based fencing mode only with CP servers as coordination points.

See “Setting up server-based I/O fencing manually” on page 162.

See “Setting up majority-based I/O fencing manually ” on page 182.

2 Make sure that the VCS cluster is online and check that the fencing mode is customized mode or majority mode.

```
# vxfenadm -d
```

3 Make sure that the cluster attribute UseFence is set to SCSI-3.

```
# haclus -value UseFence
```

4 On each node, edit the /etc/vxenviron file as follows:

```
data_disk_fencing=off
```

5. On each node, edit the /etc/sysconfig/vxfen file as follows:

   ```
   vxfen_vxfnd_tmt=25
   ```

6. On each node, edit the /etc/vxfenmode file as follows:

   ```
   loser_exit_delay=55
   vxfen_script_timeout=25
   ```

   Refer to the sample /etc/vxfenmode file.

7. On each node, set the value of the LLT sendhbcap timer parameter value as follows:

   - Run the following command:

     ```
     lltconfig -T sendhbcap:3000
     ```

   - Add the following line to the /etc/llttab file so that the changes remain persistent after any reboot:

     ```
     set-timer senhbcap:3000
     ```

8. On any one node, edit the VCS configuration file as follows:

   - Make the VCS configuration file writable:

     ```
     # haconf -makerw
     ```

   - For each resource of the type DiskGroup, set the value of the MonitorReservation attribute to 0 and the value of the Reservation attribute to NONE.

     ```
     # hares -modify <dg_resource> MonitorReservation 0

     # hares -modify <dg_resource> Reservation "NONE"
     ```

   - Run the following command to verify the value:

     ```
     # hares -list Type=DiskGroup MonitorReservation!=0

     # hares -list Type=DiskGroup Reservation!="NONE"
     ```

     The command should not list any resources.

   - Modify the default value of the Reservation attribute at type-level.

     ```
     # haattr -default DiskGroup Reservation "NONE"
     ```

- Make the VCS configuration file read-only

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

9. Make sure that the UseFence attribute in the VCS configuration file main.cf is set to SCSI-3.

10. To make these VxFEN changes take effect, stop and restart VxFEN and the dependent modules

    - On each node, run the following command to stop VCS:
      For RHEL 7, SLES 12, and supported RHEL distributions:

      ```
      # systemctl stop vcs
      ```

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

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

    - After VCS takes all services offline, run the following command to stop VxFEN:
      For RHEL 7, SLES 12, and supported RHEL distributions:

      ```
      # systemctl stop vxfen
      ```

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

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

    - On each node, run the following commands to restart VxFEN and VCS:
      For RHEL 7, SLES 12, and supported RHEL distributions:

      ```
      # systemctl start vxfen
      # systemctl start vcs
      ```

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

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

Sample /etc/vxfenmode file for non-SCSI-3 fencing

```
#
# vxfen_mode determines in what mode VCS I/O Fencing should work.
#
# available options:
# scsi3      - use scsi3 persistent reservation disks
# customized - use script based customized fencing
# disabled   - run the driver but don't do any actual fencing
#
vxfen_mode=customized
```

```
# vxfen_mechanism determines the mechanism for customized I/O
# fencing that should be used.
#
# available options:
# cps      - use a coordination point server with optional script
#            controlled scsi3 disks
#
vxfen_mechanism=cps

#
# scsi3_disk_policy determines the way in which I/O fencing
# communicates with the coordination disks. This field is
# required only if customized coordinator disks are being used.
#
# available options:
# dmp - use dynamic multipathing
#
scsi3_disk_policy=dmp

#
# Seconds for which the winning sub cluster waits to allow for the
# losing subcluster to panic & drain I/Os. Useful in the absence of
# SCSI3 based data disk fencing loser_exit_delay=55
#
# Seconds for which vxfend process wait for a customized fencing
# script to complete. Only used with vxfen_mode=customized
# vxfen_script_timeout=25

#
# vxfen_honor_cp_order determines the order in which vxfen
# should use the coordination points specified in this file.
#
# available options:
# 0 - vxfen uses a sorted list of coordination points specified
# in this file, the order in which coordination points are specified
# does not matter.
#   (default)
# 1 - vxfen uses the coordination points in the same order they are
#    specified in this file

# Specify 3 or more odd number of coordination points in this file,
# each one in its own line. They can be all-CP servers, all-SCSI-3
# compliant coordinator disks, or a combination of CP servers and
# SCSI-3 compliant coordinator disks.
# Please ensure that the CP server coordination points are
# numbered sequentially and in the same order on all the cluster
# nodes.
#
# Coordination Point Server(CPS) is specified as follows:
#
#  cps<number>=[<vip/vhn>]:<port>
#
# If a CPS supports multiple virtual IPs or virtual hostnames
# over different subnets, all of the IPs/names can be specified
# in a comma separated list as follows:
#
# cps<number>=[<vip_1/vhn_1>]:<port_1>,[<vip_2/vhn_2>]:<port_2>,
# ...,[<vip_n/vhn_n>]:<port_n>
#
# Where,
# <number>
#  is the serial number of the CPS as a coordination point; must
#  start with 1.
# <vip>
#  is the virtual IP address of the CPS, must be specified in
#  square brackets ("[]").
# <vhn>
#  is the virtual hostname of the CPS, must be specified in square
#  brackets ("[]").
# <port>
#  is the port number bound to a particular <vip/vhn> of the CPS.
#  It is optional to specify a <port>. However, if specified, it
#  must follow a colon (":") after <vip/vhn>. If not specified, the
#  colon (":") must not exist after <vip/vhn>.
#
# For all the <vip/vhn>s which do not have a specified <port>,
# a default port can be specified as follows:
#
# port=<default_port>
#
#  Where <default_port> is applicable to all the <vip/vhn>s for which a
#  <port> is not specified. In other words, specifying <port> with a
#  <vip/vhn> overrides the <default_port> for that <vip/vhn>.
# If the <default_port> is not specified, and there are <vip/vhn>s for
# which <port> is not specified, then port number 14250 will be used
# for such <vip/vhn>s.
#
# Example of specifying CP Servers to be used as coordination points:
# port=57777
# cps1=[192.168.0.23],[192.168.0.24]:58888,[cps1.company.com]
#  cps2=[192.168.0.25]
# cps3=[cps2.company.com]:59999
#
# In the above example,
# - port 58888 will be used for vip [192.168.0.24]
# - port 59999 will be used for vhn [cps2.company.com], and
# - default port 57777 will be used for all remaining <vip/vhn>s:
#    [192.168.0.23]
#    [cps1.company.com]
#    [192.168.0.25]
# - if default port 57777 were not specified, port 14250 would be
#  used for all remaining <vip/vhn>s:
#    [192.168.0.23]
#    [cps1.company.com]
#    [192.168.0.25]
#
# SCSI-3 compliant coordinator disks are specified as:
#
#  vxfendg=<coordinator disk group name>
# Example:
#  vxfendg=vxfencoorddg
#
# Examples of different configurations:
#  1. All CP server coordination points
# cps1=
# cps2=
# cps3=
#
# 2. A combination of CP server and a disk group having two SCSI-3
# coordinator disks
# cps1=
# vxfendg=
# Note: The disk group specified in this case should have two disks
#
# 3. All SCSI-3 coordinator disks
# vxfendg=
# Note: The disk group specified in case should have three disks
# cps1=[cps1.company.com]
# cps2=[cps2.company.com]
# cps3=[cps3.company.com]
# port=443
```

Setting up majority-based I/O fencing manually

Table 8-5 lists the tasks that are involved in setting up I/O fencing.

Task	Reference
Creating I/O fencing configuration files	Creating I/O fencing configuration files
Modifying VCS configuration to use I/O fencing	Modifying VCS configuration to use I/O fencing
Verifying I/O fencing configuration	Verifying I/O fencing configuration

Creating I/O fencing configuration files

To update the I/O fencing files and start I/O fencing

1. On all cluster nodes, run the following command

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

2. To check the updated /etc/vxfenmode configuration, enter the following command on one of the nodes.

   ```
   # cat /etc/vxfenmode
   ```

3. Ensure that you edit the following file on each node in the cluster to change the values of the `VXFEN_START` and the `VXFEN_STOP` environment variables to 1.

   ```
   /etc/sysconfig/vxfen
   ```

Modifying VCS configuration to use I/O fencing

After you configure I/O fencing, add the UseFence = SCSI3 cluster attribute to the VCS configuration file /etc/VRTSvcs/conf/config/main.cf.

If you reset this attribute to UseFence = None, VCS does not make use of I/O fencing abilities while failing over service groups. However, I/O fencing needs to be disabled separately.

To modify VCS configuration to enable I/O fencing

1. Save the existing configuration:

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

2. Stop VCS on all nodes:

   ```
   # hastop -all
   ```

3. To ensure High Availability has stopped cleanly, run `gabconfig -a`.

   In the output of the commans, check that Port h is not present.

4. If the I/O fencing driver vxfen is already running, stop the I/O fencing driver.

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

   ```
   # systemctl stop vxfen
   ```

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

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

5. Make a backup of the main.cf file on all the nodes:

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

6. On one node, use vi or another text editor to edit the main.cf file. To modify the list of cluster attributes, add the UseFence attribute and assign its value as SCSI3.

   ```
   cluster clus1(
   UserNames = { admin = "cDRpdxPmHpzS." }
   Administrators = { admin }
   HacliUserLevel = COMMANDROOT
   CounterInterval = 5
   UseFence = SCSI3
   )
   ```

   For fencing configuration in any mode except the disabled mode, the value of the cluster-level attribute UseFence is set to SCSI3.

7. Save and close the file.

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

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

9. Using rcp or another utility, copy the VCS configuration file from a node (for example, sys1) to the remaining cluster nodes.

   For example, on each remaining node, enter:

   ```
   # rcp sys1:/etc/VRTSvcs/conf/config/main.cf \
   /etc/VRTSvcs/conf/config
   ```

10. Start the I/O fencing driver and VCS. Perform the following steps on each node:

    - Start the I/O fencing driver.
      The vxfen startup script also invokes the `vxfenconfig` command, which configures the vxfen driver.
      For RHEL 7, SLES 12, and supported RHEL distributions:

      ```
      # systemctl start vxfen
      ```

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

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

    - Start VCS on the node where main.cf is modified.

      ```
      # /opt/VRTS/bin/hastart
      ```

    - Start VCS on all other nodes once VCS on first node reaches RUNNING state.

      ```
      # /opt/VRTS/bin/hastart
      ```

Verifying I/O fencing configuration

Verify from the vxfenadm output that the fencing mode reflects the configuration in the /etc/vxfenmode file.

To verify I/O fencing configuration

- On one of the nodes, type:

  ```
  # vxfenadm -d
  ```

  Output similar to the following appears if the fencing mode is majority:

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

   Fencing Protocol Version: 201
   Fencing Mode: MAJORITY
   Cluster Members:

          * 0 (sys1)
            1 (sys2)

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

Sample /etc/vxfenmode file for majority-based fencing

```
#
# vxfen_mode determines in what mode VCS I/O Fencing should work.
#
# available options:
# scsi3      - use scsi3 persistent reservation disks
# customized - use script based customized fencing
# majority   - use majority based fencing
# disabled   - run the driver but don't do any actual fencing
#
# vxfen_mode=majority
```

Section 4

Upgrading VCS

Chapter 9

Planning to upgrade VCS

This chapter includes the following topics:

About upgrading to VCS 7.4.2

When you upgrade to VCS 7.4.2, you need not reconfigure application monitoring with VCS. All existing monitoring configurations are preserved.

You can upgrade VCS using one of the following methods:

- Typical upgrade using product installer
  See "Upgrading VCS using the product installer" on page 196.
- Performing an online upgrade
  Perform a script-based online upgrade of your installation to upgrade VCS without stopping your applications. The supported upgrade paths for the online upgrades are same as those documented under the script and web-based upgrades.

See “Upgrading VCS online using the installer” on page 203.

- Automated upgrade using response files
  See “Upgrading VCS using response files” on page 225.

You can upgrade VCS to InfoScale Availability 7.4.2 using the product installer or response files.

See the *Veritas InfoScale Installation Guide*.

Note: In a VMware virtual environment, you can use the vSphere Client to directly install VCS and supported high availability agents (together called guest components) on the guest virtual machines. For details, see the *High Availability Solution Guide for VMware*.

You must configure the Veritas Telemetry Collector while upgrading, if you have do not already have it configured. For more information, refer to the *About telemetry data collection in InfoScale* section in the *Veritas Installation guide*.

Upgrading VCS in secure enterprise environments

In secure enterprise environments, ssh or rsh communication is not allowed between systems. In such cases, the installer program can upgrade VCS only on systems with which it can communicate (most often the local system only).

To upgrade VCS in secure enterprise environments with no rsh or ssh communication

1. Run the installer program on each node to upgrade the cluster to VCS 7.4.2.

   On each node, the installer program updates the configuration, stops the cluster, and then upgrades VCS on the node. The program also generates a cluster UUID on the node. Each node may have a different cluster UUID at this point.

2. Start VCS on the first node.

   ```
   # hastart
   ```

   VCS generates the cluster UUID on this node. Run the following command to display the cluster UUID on the local node:

   ```
   # /opt/VRTSvcs/bin/uuidconfig.pl -clus -display systemname
   ```

3. On each of the other nodes, perform the following steps:

   - Set the value of the VCS_HOST environment variable to the name of the first node.

- Display the value of the CID attribute that stores the cluster UUID value:

  ```
  # haclus -value CID
  ```

- Copy the output of the CID attribute to the file /etc/vx/.uuids/clusuuid.
- Update the VCS_HOST environment variable to remove the set value.
- Start VCS.
  The node must successfully join the already running nodes in the cluster.

Supported upgrade paths

You can upgrade to Veritas InfoScale 7.4.2 only if your currently installed product has one of the base versions: 6.2.1, 7.2, 7.3.1, 7.4.1. If your existing installation does not have one of these base versions, you must first upgrade your current installation to one of these versions. Then, follow the procedures mentioned in the Configuration and Upgrade Guide for the component configured with your InfoScale product.

If you are on an unsupported operating system version, ensure that you first upgrade to a supported version of the operating system. Also, upgrades between major operating system versions are not supported, for example, from RHEL 6 to RHEL 7. If you plan to upgrade from one major operating system version to another, you need to reinstall the product. For supported operating system versions, see the *Veritas InfoScale Release Notes*.

Table 9-1 lists the supported upgrade paths for upgrades on RHEL, Oracle Linux, and SELS.

Table 9-1 Supported upgrade paths on RHEL, Oracle Linux, and SLES

From product version	From OS version	To OS version	To product version	To Component
6.2.1	RHEL 7 Update 1, 2, 3, 4, 5, 6 , 7 Oracle Linux 7 Update 2, 3, 4, 5, 6, 7 SLES 12 SP0, SP1, SP2, SP3, SP4	RHEL 7 Update7 RHEL 8 Update 1 Oracle Linux 7 Update 7 Oracle Linux 8 Update 1 SLES 12 SP4, SP5 SLES 15 SP1	Veritas InfoScale Availability 7.4.2	VCS
7.2	RHEL 7 Update 1, 2, 3, 4, 5, 6, 7 Oracle Linux 7 Update 1, 2, 3, 4, 5, 6, 7 SLES 12 SP0, SP1, SP2	RHEL 7 Update7 RHEL 8 Update 1 Oracle Linux 7 Update 7 Oracle Linux 8 Update 1 SLES 12 SP4, SP5 SLES 15 SP1	Veritas InfoScale Availability 7.4.2	VCS
7.3.1	RHEL 7 Update 3, 4, 5, 6, 7 Oracle Linux 7 Update 3, 4, 5, 6, 7 CentOS 7 Update 3, 4, 5, 6, 7 SLES 12 SP2, SP3, SP4, SP5	RHEL 7 Update7 RHEL 8 Update 1 Oracle Linux 7 Update 7 Oracle Linux 8 Update 1 CentOS 7 Update 7 CentOS 8 Update 1 SLES 12 SP4, SP5 SLES 15 SP1	Veritas InfoScale Availability 7.4.2	VCS

Table 9-1 Supported upgrade paths on RHEL, Oracle Linux, and SLES *(continued)*

From product version	From OS version	To OS version	To product version	To Component
7.4.1	RHEL 7 Update 4, 5, 6, 7 RHEL 8 update 1 Oracle Linux 7 Update 4, 5, 6, 7 CentOS 7 Update 4, 5, 6 , 7 SLES 12 SP2, SP3, SP4, SP5 SLES 15 SP1	RHEL 7 Update7 RHEL 8 Update 1 Oracle Linux 7 Update 7 Oracle Linux 8 Update 1 CentOS 7 Update 7 CentOS 8 Update 1 SLES 12 SP4, SP5 SLES 15 SP1	Veritas InfoScale Availability 7.4.2	VCS

Considerations for upgrading secure VCS 6.x clusters to VCS 7.4.2

When you upgrade a secure VCS 6.x cluster to VCS 7.4.2, the upgrade does not migrate the old broker configuration to the new broker because of the change in architecture. Both the old broker (`/opt/VRTSat/bin/vxatd`) and new broker (`/opt/VRTSvcs/bin/vcsauth/vcsauthserver/bin/vcsauthserver`) continue to run. In such a scenario, you must consider the following:

- The HA commands that you run in VCS 7.4.2 are processed by the new broker by default. To ensure that the HA commands are processed by the old broker, set the VCS_REMOTE_BROKER environment variable as follows:

  ```
  # export VCS_REMOTE_BROKER=localhost IP,2821
  ```

- VCS 7.4.2 does not prompt non-root users who run HA commands for passwords. In 6.x, non-root users required a password to run HA commands. If you want non-root users to enter passwords before they run HA commands, set the VCS_DOMAINTYPE environment variable to unixpwd.

- Trust relationships are not migrated during the upgrade. If you had configured secure GCO or secure steward, ensure that trust relationships are recreated between the clusters and the steward.
  See "Setting up trust relationships for your VCS cluster" on page 63.

When the old broker is not used anymore, you can delete the old VRTSat RPM.

Considerations for upgrading VCS to 7.4.2 on systems configured with an Oracle resource

If you plan to upgrade VCS running on systems configured with an Oracle resource, set the `MonitorOption` attribute to 0 (zero) before you start the upgrade.

For more information on enabling the Oracle health check, see the *Cluster Server Agent for Oracle Installation and Configuration Guide*.

Considerations for upgrading CP servers

From release 7.0.1 onwards, VCS does not support IPM-based (Veritas Product Authentication Services) secure communication between clusters.

The only supported communication protocol is HTTPS-based communication. For HTTPS communication, you do not need to consider setting up trust relationships between CP servers and clients.

Considerations for upgrading CP clients

Passwordless communication from CP clients to CP server must exist for the installer to reconfigure fencing. If passwordless communication does not exist, you must reconfigure fencing manually.

See "Setting up disk-based I/O fencing manually" on page 157.

See "Setting up server-based I/O fencing manually" on page 162.

Using Install Bundles to simultaneously install or upgrade full releases (base, maintenance, rolling patch), and individual patches

Beginning with version 6.2.1, you can easily install or upgrade your systems directly to a base, maintenance, patch level or a combination of multiple patches and packages together in one step using Install Bundles. With Install Bundles, the installer has the ability to merge so that customers can install or upgrade directly to maintenance or patch levels in one execution. The various scripts, RPMs, and patch components are merged, and multiple releases are installed together as if they are one combined release. You do not have to perform two or more install actions to install or upgrade systems to maintenance levels or patch levels.

Releases are divided into the following categories:

Table 9-2 Release Levels

Level	Content	Form factor	Applies to	Release types	Download location
Base	Features	RPMs	All products	Major, minor, Service Pack (SP), Platform Release (PR)	FileConnect
Maintenance	Fixes, new features	RPMs	All products	Maintenance Release (MR), Rolling Patch (RP)	Veritas Services and Operations Readiness Tools (SORT)
Patch	Fixes	RPMs	Single product	P-Patch, Private Patch, Public patch	SORT, Support site

When you install or upgrade using Install Bundles:

- Veritas InfoScale products are discovered and assigned as a single version to the maintenance level. Each system can also have one or more patches applied.
- Base releases are accessible from FileConnect that requires customer serial numbers. Maintenance and patch releases can be automatically downloaded from SORT.
- Patches can be installed using automated installers.
- Patches can now be detected to prevent upgrade conflict. Patch releases are not offered as a combined release. They are only available from Veritas Technical Support on a need basis.

You can use the `-base_path` and `-patch_path` options to import installation code from multiple releases. You can find RPMs and patches from different media paths, and merge RPM and patch definitions for multiple releases. You can use these options to use new task and phase functionality to correctly perform required operations for each release component. You can install the RPMs and patches in defined phases using these options, which helps you when you want to perform a single start or stop process and perform pre and post operations for all level in a single operation.

Four possible methods of integration exist. All commands must be executed from the highest base or maintenance level install script.

In the example below:

- 7.4.2 is the base version
- 7.4.2.1 is the maintenance version
- 7.4.2.1.100 is the patch version for 7.4.2.1
- 7.4.2.0.100 is the patch version for 7.4.2

1. Base + maintenance:

   This integration method can be used when you install or upgrade from a lower version to 7.4.2.1.

   Enter the following command:

   ```
   # installmr -base_path <path_to_base>
   ```

2. Base + patch:

   This integration method can be used when you install or upgrade from a lower version to 7.4.2.0.100.

   Enter the following command:

   ```
   # installer -patch_path <path_to_patch>
   ```

3. Maintenance + patch:

   This integration method can be used when you upgrade from version 7.4.2 to 7.4.2.1.100.

   Enter the following command:

   ```
   # installmr -patch_path <path_to_patch>
   ```

4. Base + maintenance + patch:

   This integration method can be used when you install or upgrade from a lower version to 7.4.2.1.100.

   Enter the following command:

   ```
   # installmr -base_path <path_to_base>
   -patch_path <path_to_patch>
   ```

Note: You can add a maximum of five patches using *-patch_path <path_to_patch> -patch2_path <path_to_patch> ... -patch5_path <path_to_patch>*

Chapter 10

Performing a VCS upgrade using the installer

This chapter includes the following topics:

Before upgrading VCS using the script-based installer

As a result of OS upgrade, if VCS is not in running state before upgrade, the installer does not start VCS after the upgrade is completed. You need to manually start it or restart the cluster nodes. Before you upgrade VCS, you first need to remove deprecated resource types and modify changed values.

Upgrading VCS using the product installer

You can use the product installer to upgrade VCS.

To upgrade VCS using the product installer

1. Log in as superuser and mount the product disc.
2. Start the installer.

   ```
   # ./installer
   ```

   The installer starts the product installation program with a copyright message. It then specifies where it creates the logs. Note the log's directory and name.
3. Select the product you want to upgrade.
4. Choose **1** for full upgrade.
5. Enter the names of the nodes that you want to upgrade. Use spaces to separate node names. Press the Enter key to proceed.
6. When the verification checks are complete, the installer asks if you agree with the terms of the End User License Agreement. Press **y** to agree and continue.
7. The installer displays the following question before the install stops the product processes. If the cluster was not configured in secure mode before the upgrade, these questions are not displayed.

   - Do you want to generate certificates for the Veritas InfoScale Enterprise components with 2048 bit encryption keys and SHA256 signatures? [y,n,q,?]
     You can enter **y** and upgrade to 2048 bit key and SHA256 signature certificates at this stage or perform the upgrade later.
8. The installer asks if you want to stop VCS processes. Press the Enter key to continue.

   The installer stops VCS processes, uninstalls RPMs, installs or upgrades RPMs, and configures VCS.

   The installer lists the nodes that Veritas recommends you to restart.
9. The installer asks if you want to send the information about this installation to us to help improve installation in the future. Enter your response.

   The installer displays the location of log files, summary file, and response file.
10. If you want to upgrade CP server systems that use VCS or SFHA to VCS 7.4.2, make sure that you first upgrade all application clusters to version VCS 7.4.2. Then, upgrade VCS or SFHA on the CP server systems.

    For instructions to upgrade SFHA, see the *Storage Foundation and High Availability Configuration and Upgrade Guide*.

If you did not upgrade to 2048 bit key and SHA256 signature certificates in the above procedure, refer to *Upgrading to 2048 bit key and SHA256 signature certificates* when you upgrade the certificates later.

See “Upgrading to 2048 bit key and SHA256 signature certificates” on page 198.

Upgrading to 2048 bit key and SHA256 signature certificates

Perform the following steps to upgrade to 2048 bit key and SHA256 signature certificates:

1. Run the following command to begin the upgrade:

   ```
   # /opt/VRTS/install/installer -security
   ```

   The system displays the following message:

   ```
   Would you like to configure secure mode on the cluster? [y,n,q]
   ```

2. Enter **y**. The system displays the message that you have to re-establish the trust relationships between components after upgrading the certificates.
3. Enter **1** to begin the upgrade.
4. The system displays the following message after the security upgrade:

   ```
   Security reconfiguration completed successfully.
   ```

   You can optionally view the upgrade logs.

Tasks to perform after upgrading to 2048 bit key and SHA256 signature certificates

Note that you must perform the following tasks after upgrading to 2048 bit key and SHA256 signature certificates:

- Delete certificates of non-root users after upgrading to 2048 bit key and SHA256 signature certificates.
  See “Deleting certificates of non-root users after upgrading to 2048 bit key and SHA256 signature certificates” on page 199.
- Re-establish WAC communication in global clusters after upgrading to 2048 bit key and SHA256 signature certificates.
  See “Re-establishing WAC communication in global clusters after upgrading to 2048 bit key and SHA256 signature certificates” on page 199.
- Re-establish communication between Steward and upgraded clusters.

See "Re-establishing trust with Steward after upgrading to 2048 bit key and SHA256 signature certificates" on page 201.

Note: The cluster communication between global clusters breaks if the cluster on one site is running VCS 7.4.2 and the other is on VCS version lower than 6.0.5. If you upgrade to 2048 bit key and SHA256 signature certificates in such a configuration, the communication will not be restored even after performing the essential tasks after the upgrade. The only workaround for this is to upgrade VCS to version 6.0.5 or above on clusters which are running on VCS versions lower than 6.0.5.

Deleting certificates of non-root users after upgrading to 2048 bit key and SHA256 signature certificates

After upgrading to 2048 bit key and SHA256 signature certificates, the certificates of non-root users become obsolete and must be deleted in order to allow non-root users to log in. Perform the following steps to delete the non-root certificates for each non-root user:

1. Delete .VRTSat from the home directory of the users.

   ```
   # rm -rf /user_home_directory/.VRTSat
   ```

2. If the non-root user was not a cluster user prior to upgrade, add user to Cluster with appropriate Privilege.

   ```
   # hauser -add <user name> -priv <Privilege>
   ```

3. Log in with non-root user and create new certificates.

   ```
   # /opt/VRTS/bin/halogin <non_root_user> <password>
   ```

Re-establishing WAC communication in global clusters after upgrading to 2048 bit key and SHA256 signature certificates

During the upgrade, the vcsauthserver gets 2048 bit SHA256 certificates and the trust information gets deleted, which causes the WAC communication to break. To establish the communication again, you must set up trust for WAC on each node of every cluster. The remote site has to set up trust with the local site as a new broker certificate is created on the local site. The local site also has to set up trust with the remote site as the trust certificate gets deleted during the upgrade.

Perform the following steps to establish trust between the clusters:

1 On each node of the first cluster, run the following command:

```
# export EAT_DATA_DIR=/var/VRTSvcs/vcsauth/data/WAC;
/opt/VRTSvcs/bin/vcsat setuptrust -b
IP_address_of_any_node_from_the_second_cluster:14149 -s high
```

The command obtains and displays the security certificate and other details of the root broker of the second cluster. If the details are correct, enter **y** at the command prompt to establish trust.

For example: The hash of above credential is b36a2607bf48296063068e3fc49188596aa079bb

```
Do you want to trust the above?(y/n) y
```

2 On each node of the second cluster, run the following command:

```
# export EAT_DATA_DIR=/var/VRTSvcs/vcsauth/data/WAC;
/opt/VRTSvcs/bin/vcsat setuptrust -b
IP_address_of_any_node_from_the_first_cluster:14149 -s high
```

The command obtains and displays the security certificate and other details of the root broker of the first cluster. If the details are correct, enter **y** at the command prompt to establish trust.

Re-establishing CP server and CP client communication after upgrading to 2048 bit key and SHA256 signature certificates

In case of CP server, when you upgrade the CP server cluster to use the enhanced security, the vcsauthserver gets 2048 bit SHA256 certificates and the trust information is no longer valid on the client clusters. This breaks the communication between CP server and CP client. To reinstate the communication, you must setup trust between CP server and CP client.

For each client node on VCS 6.0 and later, run the following command:

```
EAT_DATA_DIR=/var/VRTSvcs/vcsauth/data/CPSADM \
/opt/VRTSvcs/bin/vcsat setuptrust -b cpserver_ip_address:14149 -s high
```

Re-establishing trust with Steward after upgrading to 2048 bit key and SHA256 signature certificates

In case of Steward, when you upgrade the cluster to use the enhanced security, the vcsauthserver gets 2048 bit SHA256 certificates and the trust information gets deleted. This breaks the communication between cluster and Steward.

To reinstate the communication, you must setup trust between each node of the upgraded cluster and Steward.

1 Set up trust on all nodes of the GCO clusters:

```
# export EAT_DATA_DIR=/var/VRTSvcs/vcsauth/data/WAC
# vcsat setuptrust -b IP_of_Steward:14149 -s high
```

2 Set up trust on the Steward:

```
# export EAT_DATA_DIR=/var/VRTSvcs/vcsauth/data/STEWARD
# vcsat setuptrust -b VIP_of_upgraded_cluster:14149 -s high
```

Upgrading Steward to 2048 bit key and SHA256 signature certificates

To upgrade Steward after upgrading to 2048 bit key and SHA256 signature certificates:

1 Log on to the Steward system as a root user.

2 Stop the Steward process.

```
# steward -stop -secure
```

3 Remove `/var/VRTSvcs/vcsauth/data/STEWARD`.

```
# rm -rf /var/VRTSvcs/vcsauth/data/STEWARD
```

4 Upgrade the VRTSvcs and VRTSperl RPMs using:

```
# rpm -Uvh
```

5 Run `./installer -securityonenode`

The installer prompts for a confirmation if VCS is not configured or if VCS is not running on all nodes of the cluster.

6 Enter **y** when the installer prompts whether you want to continue configuring security.

7 Run `/opt/VRTSvcs/bin/steward_secure.pl`.

8 Set up trust on all nodes of the GCO clusters:

```
# export EAT_DATA_DIR=/var/VRTSvcs/vcsauth/data/WAC
# vcsat setuptrust -b IP_of_Steward:14149 -s high
```

9 Set up trust on the Steward for every GCO cluster:

```
# export EAT_DATA_DIR=/var/VRTSvcs/vcsauth/data/STEWARD
# vcsat setuptrust -b VIP_of_remote_cluster1:14149 -s high
# vcsat setuptrust -b VIP_of_remote_cluster2:14149 -s high.
```

Performing an online upgrade

This chapter includes the following topics:

- Limitations of online upgrade
- Upgrading VCS online using the installer

Limitations of online upgrade

- Online upgrade is available only for VCS. If you have Storage Foundation, SFHA, SFCFSHA, or any other solution with VxVM and VxFS installed, then the online upgrade process is not supported.
- The non-Veritas applications running on the node have zero down time during the online upgrade.
- VCS does not monitor the applications when online upgrade is in progress.
- For upgrades from VCS versions lower than 6.1, upgrade the CP server before performing the online upgrade.

See "Upgrading VCS online using the installer" on page 203.

Upgrading VCS online using the installer

You can use the product installer to upgrade VCS online. The supported upgrade paths are same as those for the installer.

To upgrade VCS online using the product installer

1. Log in as superuser and mount the product disc.
2. Start the installer.

   ```
   # ./installer
   ```

   The installer starts the product installation program with a copyright message. It then specifies where it creates the logs.

   Note the directory name and path where the logs get stored.
3. Choose **Online Upgrade** from the upgrade options.
4. After selecting the online upgrade method, enter any one system name from the cluster on which you want to perform the online upgrade.

   Even if you specify a single node from the cluster, the installer asks whether you want to perform online upgrade of VCS on the entire cluster, keeping your applications online. After you enter the system name, the installer performs some verification checks and asks the following question:

   ```
   Online upgrade supports application zero downtime.
   Would you like to perform online upgrade on the
   whole cluster? [y,n,q](y)
   ```
5. Enter **y** to initiate the online upgrade.

   Note: You can either exit the installer with the option **q** or cancel the upgrade using **n** and select any other cluster to upgrade in this step.

   The installer runs some verification checks on the nodes and subsequently asks if you agree with the terms of the End User License Agreement.
6. Enter **y** to agree and continue.
7. The installer displays the following question before the installer stops the product processes. If the cluster was configured in secure mode and the version is prior to 6.2 before upgrade, these questions will be displayed.
   - Do you want to grant read access to everyone? [y,n,q,?]
     - To grant read access to all authenticated users, type **y**.
     - To grant usergroup specific permissions, type **n**.
   - Do you want to provide any usergroups that you would like to grant read access?[y,n,q,?]
     - To specify usergroups and grant them read access, type **y**

    - To grant read access only to root users, type **n**. The installer grants read access to the root users.

  - Enter the usergroup names separated by spaces that you want to grant read access. If you want to grant read access to a usergroup on a specific node, enter like 'usrgrp1@node1', and if you want to grant read access to usergroup on any cluster node, enter like 'usrgrp1'. If some usergroups are not created yet, create the usergroups after configuration if needed. [b]

8. The installer asks if you want to stop VCS processes. Enter **y** to stop the VCS process.

   It stops the VCS processes, uninstalls RPMs, reinstalls or upgrades RPMs, again configures VCS, and starts the processes.

9. The installer asks if you want to stop VCS processes. Enter **y** to stop the VCS process.

Chapter 12

Performing a phased upgrade of VCS

This chapter includes the following topics:

- About phased upgrade
- Performing a phased upgrade using the product installer

About phased upgrade

Perform a phased upgrade to minimize the downtime for the cluster.

Depending on the situation, you can calculate the approximate downtime as follows:

Table 12-1

Fail over condition	Downtime
You can fail over all your service groups to the nodes that are up.	Downtime equals the time that is taken to offline and online the service groups.
You have a service group that you cannot fail over to a node that runs during upgrade.	Downtime for that service group equals the time that is taken to perform an upgrade and restart the node.

Prerequisites for a phased upgrade

Before you start the upgrade, confirm that you have licenses for all the nodes that you plan to upgrade.

Planning for a phased upgrade

Plan out the movement of the service groups from node-to-node to minimize the downtime for any particular service group.

Some rough guidelines follow:

- Split the cluster into two subclusters of equal or near equal size.
- Split the cluster so that your high priority service groups remain online during the upgrade of the first subcluster.
- Before you start the upgrade, back up the VCS configuration files `main.cf` and `types.cf` which are in the `/etc/VRTSvcs/conf/config/` directory.
- Before you start the upgrade make sure that all the disk groups have the latest backup of configuration files in the `/etc/vx/cbr/bk` directory. If not, then run the following command to take the latest backup.

  ```
  # /etc/vx/bin/vxconfigbackup -| [dir] [dgname|dgid]
  ```

Phased upgrade limitations

The following limitations primarily describe not to tamper with configurations or service groups during the phased upgrade:

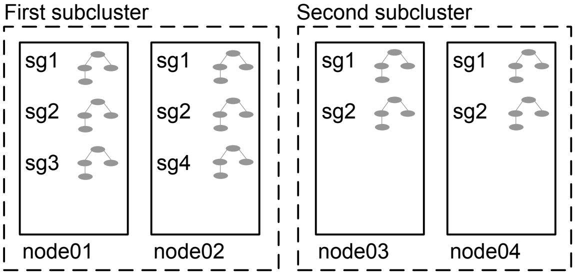
- While you perform the upgrades, do not start any modules.
- When you start the installer, only select VCS.
- While you perform the upgrades, do not add or remove service groups to any of the nodes.
- After you upgrade the first half of your cluster (the first subcluster), you need to set up password-less ssh or rsh. Create the connection between an upgraded node in the first subcluster and a node from the other subcluster. The node from the other subcluster is where you plan to run the installer and also plan to upgrade.
- Depending on your configuration, you may find that you cannot upgrade multiple nodes at the same time. You may only be able to upgrade one node at a time.
- For very large clusters, you might have to repeat these steps multiple times to upgrade your cluster.

Phased upgrade example

In this example, you have a secure cluster that you have configured to run on four nodes: node01, node02, node03, and node04. You also have four service groups:

sg1, sg2, sg3, and sg4. For the purposes of this example, the cluster is split into two subclusters. The nodes node01 and node02 are in the first subcluster, which you first upgrade. The nodes node03 and node04 are in the second subcluster, which you upgrade last.

Figure 12-1 Example of phased upgrade set up



Each service group is running on the nodes as follows:

- sg1 and sg2 are parallel service groups and run on all the nodes.
- sg3 and sg4 are failover service groups. sg3 runs on node01 and sg4 runs on node02.

In your system list, you have each service group that fails over to other nodes as follows:

- sg1 and sg2 are running on all the nodes.
- sg3 and sg4 can fail over to any of the nodes in the cluster.

Phased upgrade example overview

This example's upgrade path follows:

- Move all the failover service groups from the first subcluster to the second subcluster.
- Take all the parallel service groups offline on the first subcluster.
- Upgrade the operating system on the first subcluster's nodes, if required.
- On the first subcluster, start the upgrade using the installation program.
- Get the second subcluster ready.
- Activate the first subcluster. After activating the first cluster, switch the service groups online on the second subcluster to the first subcluster.
- Upgrade the operating system on the second subcluster's nodes, if required.
- On the second subcluster, start the upgrade using the installation program.

- Activate the second subcluster.

See “Performing a phased upgrade using the product installer” on page 209.

Performing a phased upgrade using the product installer

This section explains how to perform a phased upgrade of VCS on four nodes with four service groups. Note that in this scenario, VCS and the service groups cannot stay online on the second subcluster during the upgrade of the second subcluster. Do not add, remove, or change resources or service groups on any nodes during the upgrade. These changes are likely to get lost after the upgrade.

An example of a phased upgrade follows. It illustrates the steps to perform a phased upgrade. The example makes use of a secure VCS cluster.

You can perform a phased upgrade from VCS 6.0 to VCS 7.4.2.

See “About phased upgrade” on page 206.

See “Phased upgrade example” on page 207.

Moving the service groups to the second subcluster

Perform the following steps to establish the service group's status and to switch the service groups.

To move service groups to the second subcluster

1 On the first subcluster, determine where the service groups are online.

```
# hagrp -state
```

The output resembles:

```
#Group   Attribute System Value
sg1      State     node01 |ONLINE|
sg1      State     node02 |ONLINE|
sg1      State     node03 |ONLINE|
sg1      State     node04 |ONLINE|
sg2      State     node01 |ONLINE|
sg2      State     node02 |ONLINE|
sg2      State     node03 |ONLINE|
sg2      State     node04 |ONLINE|
sg3      State     node01 |ONLINE|
sg3      State     node02 |OFFLINE|
sg3      State     node03 |OFFLINE|
sg3      State     node04 |OFFLINE|
sg4      State     node01 |OFFLINE|
sg4      State     node02 |ONLINE|
sg4      State     node03 |OFFLINE|
sg4      State     node04 |OFFLINE|
```

2 Offline the parallel service groups (sg1 and sg2) from the first subcluster. Switch the failover service groups (sg3 and sg4) from the first subcluster (node01 and node02) to the nodes on the second subcluster (node03 and node04). For SFHA, vxfen sg is the parallel service group.

```
# hagrp -offline sg1 -sys node01
# hagrp -offline sg2 -sys node01
# hagrp -offline sg1 -sys node02
# hagrp -offline sg2 -sys node02
# hagrp -switch sg3 -to node03
# hagrp -switch sg4 -to node04
```

3 On the nodes in the first subcluster, unmount all the VxFS file systems that VCS does not manage, for example:

```
# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/sda1              26G  3.3G   22G  14% /
udev                 1007M  352K 1006M   1% /dev
tmpfs                 4.0K     0  4.0K   0% /dev/vx
/dev/vx/dsk/dg2/dg2vol1
                      3.0G   18M  2.8G   1% /mnt/dg2/dg2vol1
/dev/vx/dsk/dg2/dg2vol2
                      1.0G   18M  944M   2% /mnt/dg2/dg2vol2
/dev/vx/dsk/dg2/dg2vol3
                       10G   20M  9.4G   1% /mnt/dg2/dg2vol3
# umount /mnt/dg2/dg2vol1
# umount /mnt/dg2/dg2vol2
# umount /mnt/dg2/dg2vol3
```

4 On the nodes in the first subcluster, stop all VxVM volumes (for each disk group) that VCS does not manage.

5 Make the configuration writable on the first subcluster.

```
# haconf -makerw
```

6 Freeze the nodes in the first subcluster.

```
# hasys -freeze -persistent node01
# hasys -freeze -persistent node02
```

7 Dump the configuration and make it read-only.

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

8 Verify that the service groups are offline on the first subcluster that you want to upgrade.

```
# hagrp -state
```

Output resembles:

```
#Group Attribute System Value
sg1  State node01 |OFFLINE|
sg1  State node02 |OFFLINE|
sg1  State node03 |ONLINE|
sg1  State node04 |ONLINE|
sg2  State node01 |OFFLINE|
sg2  State node02 |OFFLINE|
sg2  State node03 |ONLINE|
sg2  State node04 |ONLINE|
sg3  State node01 |OFFLINE|
sg3  State node02 |OFFLINE|
sg3  State node03 |ONLINE|
sg3  State node04 |OFFLINE|
sg4  State node01 |OFFLINE|
sg4  State node02 |OFFLINE|
sg4  State node03 |OFFLINE|
sg4  State node04 |ONLINE|
```

Upgrading the operating system on the first subcluster

You can perform the operating system upgrade on the first subcluster, if required.

Before performing operating system upgrade, it is better to prevent LLT from starting automatically when the node starts. For example, you can do the following:

```
# mv /etc/llttab /etc/llttab.save
```

or you can change the `/etc/default/llt` file by setting LLT_START = **0**.

After you finish upgrading the OS, remember to change the LLT configuration to its original configuration.

Refer to the operating system's documentation for more information.

Upgrading the first subcluster

You now navigate to the installer program and start it.

To start the installer for the phased upgrade

1 Confirm that you are logged on as the superuser and you mounted the product disc.

2 Make sure that you can ssh or rsh from the node where you launched the installer to the nodes in the second subcluster without requests for a password.

3 Navigate to the folder that contains installvcs.

```
# cd cluster_server
```

4 Start the installer program, specify the nodes in the first subcluster (node1 and node2).

```
# ./installer node1 node2
```

The program starts with a copyright message and specifies the directory where it creates the logs.

5 Select **G (Upgrade a Product)** from Task Menu.

```
Task Menu:

    P) Perform a Pre-Installation Check     I) Install a Product
    C) Configure a Product Component        G) Upgrade a Product
    O) Perform a Post-Installation Check    U) Uninstall a Product
    L) License a Product                    S) Start a Product
    D) View Product Descriptions            X) Stop a Product
    R) View Product Requirements            ?) Help

Enter a Task: [P,I,C,G,O,U,L,S,D,X,R,?] g
```

6 Review the installation options, and select 1 (Full Upgrade).

```
   1)  Full Upgrade
   2)  Rolling Upgrade
   3)  Online Upgrade [VCS only]
   b)  Back to previous menu

Select the method by which you want to upgrade the product: [1-3,b,q](1)1
```

7. The installer performs a series of checks and tests to ensure communications, licensing, and compatibility.

8. When you are prompted, reply **y** to continue with the upgrade.

```
Do you want to continue? [y,n,q] (y)
```

9. The Enter **y** to agree to the End User License Agreement (EULA).

```
Do you agree with the terms of the End User License Agreement as
specified in the EULA/en/EULA.pdf file present on media? [y,n,q,?] y
```

10. When you are prompted, reply **y** to stop appropriate processes.

```
Do you want to stop Enterprise processes now? [y,n,q] (y)
```

The installer ends for the first subcluster with the following output:

```
Configuring VCS: 100%

Estimated time remaining: (mm:ss) 0:00                      1 of 1

Performing VCS upgrade configuration ...........................Done

Cluster Server Configure completed successfully

You are performing phased upgrade (Phase 1) on the systems.
Follow  the steps in Configuration and Upgrade Guide to upgrade the
remaining systems.

Installation procedures and diagnostic information are saved in the log
files under directory /opt/VRTStmp/[log_dir]. This information helps us
identify and resolve failed operations performed by the installer.
Would you like to send the information to us to help improve installation
in the future? [y,n,q,?] (y) y
```

The upgrade is finished on the first subcluster. Do not reboot the nodes in the first subcluster until you complete the Preparing the second subcluster procedure.

11. In the `/etc/default/llt` file, set LLT_START = 0.

Preparing the second subcluster

Perform the following steps on the second subcluster before rebooting nodes in the first subcluster.

To prepare to upgrade the second subcluster

1 Get the summary of the status of your resources.

```
# hastatus -summ
-- SYSTEM STATE
-- System               State                Frozen

A  node01              EXITED                 1
A  node02              EXITED                 1
A  node03              RUNNING                0
A  node04              RUNNING                0

-- GROUP STATE
-- Group           System  Probed     AutoDisabled    State

B  SG1             node01   Y          N               OFFLINE
B  SG1             node02   Y          N               OFFLINE
B  SG1             node03   Y          N               ONLINE
B  SG1             node04   Y          N               ONLINE
B  SG2             node01   Y          N               OFFLINE
B  SG2             node02   Y          N               OFFLINE
B  SG2             node03   Y          N               ONLINE
B  SG2             node04   Y          N               ONLINE
B  SG3             node01   Y          N               OFFLINE
B  SG3             node02   Y          N               OFFLINE
B  SG3             node03   Y          N               ONLINE
B  SG3             node04   Y          N               OFFLINE
B  SG4             node01   Y          N               OFFLINE
B  SG4             node02   Y          N               OFFLINE
B  SG4             node03   Y          N               OFFLINE
B  SG4             node04   Y          N               ONLINE
```

2 Unmount all the VxFS file systems that VCS does not manage, for example:

```
# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/sda1              26G  3.3G   22G  14% /
udev                 1007M  352K 1006M   1% /dev
tmpfs                 4.0K     0  4.0K   0% /dev/vx
/dev/vx/dsk/dg2/dg2vol1
                      3.0G   18M  2.8G   1% /mnt/dg2/dg2vol1
/dev/vx/dsk/dg2/dg2vol2
                      1.0G   18M  944M   2% /mnt/dg2/dg2vol2
/dev/vx/dsk/dg2/dg2vol3
                       10G   20M  9.4G   1% /mnt/dg2/dg2vol3
# umount /mnt/dg2/dg2vol1
# umount /mnt/dg2/dg2vol2
# umount /mnt/dg2/dg2vol3
```

3 Take the service groups offline on node03 and node04.

```
# hagrp -offline sg1 -sys node03
# hagrp -offline sg1 -sys node04
# hagrp -offline sg2 -sys node03
# hagrp -offline sg2 -sys node04
# hagrp -offline sg3 -sys node03
# hagrp -offline sg4 -sys node04
```

4 Verify the state of the service groups.

```
# hagrp -state
#Group       Attribute    System     Value
SG1            State      node01    |OFFLINE|
SG1            State      node02    |OFFLINE|
SG1            State      node03    |OFFLINE|
SG1            State      node04    |OFFLINE|
SG2            State      node01    |OFFLINE|
SG2            State      node02    |OFFLINE|
SG2            State      node03    |OFFLINE|
SG2            State      node04    |OFFLINE|
SG3            State      node01    |OFFLINE|
SG3            State      node02    |OFFLINE|
SG3            State      node03    |OFFLINE|
SG3            State      node04    |OFFLINE|
```

5 Stop all VxVM volumes (for each disk group) that VCS does not manage.

6 Stop VCS, I/O Fencing, GAB, and LLT on node03 and node04.

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

```
# hastop -local
# systemctl stop vxfen
# systemctl stop gab
# systemctl stop llt
```

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

```
# hastop -local
# /etc/init.d/vxfen stop
# /etc/init.d/gab stop
# /etc/init.d/llt stop
```

7 Make sure that the VXFEN, GAB, and LLT modules on node03 and node04 are not added.

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

```
# /opt/VRTSvcs/vxfen/bin/vxfen status
VXFEN module is not loaded

# /opt/VRTSgab/gab status
GAB module is not loaded

# /opt/VRTSllt/llt status
LLT module is not loaded
```

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

```
# /etc/init.d/vxfen status
VXFEN module is not loaded

# /etc/init.d/gab status
GAB module is not loaded

# /etc/init.d/llt status
LLT module is not loaded
```

Activating the first subcluster

Get the first subcluster ready for the service groups.

Note: These steps fulfill part of the installer's output instructions.

See "Upgrading the first subcluster" on page 213.

To activate the first subcluster

1 Start LLT and GAB on one node in the first half of the cluster.

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

```
# systemctl start llt
# systemctl start gab
```

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

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

2 Seed node01 in the first subcluster.

```
# gabconfig -x
```

3 On the first half of the cluster, start VCS:

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

```
# ./installer -start sys1 sys2
```

4 Make the configuration writable on the first subcluster.

```
# haconf -makerw
```

5 Unfreeze the nodes in the first subcluster.

```
# hasys -unfreeze -persistent node01
# hasys -unfreeze -persistent node02
```

6 Dump the configuration and make it read-only.

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

7 Bring the service groups online on node01 and node02.

```
# hagrp -online sg1 -sys node01
# hagrp -online sg1 -sys node02
# hagrp -online sg2 -sys node01
# hagrp -online sg2 -sys node02
# hagrp -online sg3 -sys node01
# hagrp -online sg4 -sys node02
```

Upgrading the operating system on the second subcluster

You can perform the operating system upgrade on the second subcluster, if required.

Before performing operating system upgrade, it is better to prevent LLT from starting automatically when the node starts. For example, you can do the following:

```
# mv /etc/llttab /etc/llttab.save
```

or you can change the `/etc/default/llt` file by setting LLT_START = **0**.

After you finish upgrading the OS, remember to change the LLT configuration to its original configuration.

Refer to the operating system's documentation for more information.

Upgrading the second subcluster

Perform the following procedure to upgrade the second subcluster (node03 and node04).

To start the installer to upgrade the second subcluster

1 Confirm that you are logged on as the superuser and you mounted the product disc.

2 Navigate to the folder that contains installvcs.

```
# cd cluster_server
```

3 Confirm that VCS is stopped on node03 and node04. Start the installvcs program, specify the nodes in the second subcluster (node3 and node4).

```
# ./installer node3 node4
```

The program starts with a copyright message and specifies the directory where it creates the logs.

4 Select **G (Upgrade a Product)** from the Task Menu.

```
Task Menu:


    P) Perform a Pre-Installation Check     I) Install a Product
    C) Configure a Product Component        G) Upgrade a Product
    O) Perform a Post-Installation Check    U) Uninstall a Product
    L) License a Product                    S) Start a Product
    D) View Product Descriptions            X) Stop a Product
    R) View Product Requirements            ?) Help

    Enter a Task: [P,I,C,G,O,U,L,S,D,X,R,?] g
```

5 Review the installation options, and select **1 (Full Upgrade)**.

```
   1)  Full Upgrade
   2)  Rolling Upgrade
   3)  Online Upgrade [VCS only]
   b)  Back to previous menu

 Select the method by which you want to upgrade the product: [1-3,b,q] (1
```

6 The installer performs a series of checks and tests to ensure communications, licensing, and compatibility.

7 When you are prompted, reply **y** to continue with the upgrade.

```
Do you want to continue? [y,n,q] (y)
```

8. Enter **y** to agree to the End User License Agreement (EULA).

   ```
   Do you agree with the terms of the End User License Agreement as
   specified in the EULA/en/EULA.pdf file present on media? [y,n,q,?] y

   When you are prompted, reply y to stop appropriate processes.
   Do you want to stop InfoScale Enterprise processes now? [y,n,q,?] (y) y
   ```

9. Monitor the installer program answering questions as appropriate until the upgrade completes.

Finishing the phased upgrade

Complete the following procedure to complete the upgrade.

To finish the upgrade

1. Verify that the cluster UUID is the same on the nodes in the second subcluster and the first subcluster. Run the following command to display the cluster UUID:

   ```
   # /opt/VRTSvcs/bin/uuidconfig.pl
   -clus -display node1 [node2 ...]
   ```

   If the cluster UUID differs, manually copy the cluster UUID from a node in the first subcluster to the nodes in the second subcluster. For example:

   ```
   # /opt/VRTSvcs/bin/uuidconfig.pl [-rsh] -clus
   -copy -from_sys node01 -to_sys node03 node04
   ```

2. On the second half of the cluster, start VCS:

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

   ```
   # ./installer -start sys3 sys4
   ```

3. Check to see if VCS and its components are up.

   ```
   # gabconfig -a
   GAB Port Memberships
   ===============================================================
   Port a gen   nxxxnn membership 0123
   Port b gen   nxxxnn membership 0123
   Port h gen   nxxxnn membership 0123
   ```

4 Run an `hastatus -sum` command to determine the status of the nodes, service groups, and cluster.

```
# hastatus -sum

-- SYSTEM STATE
-- System          State           Frozen

A  node01          RUNNING         0
A  node02          RUNNING         0
A  node03          RUNNING         0
A  node04          RUNNING         0

-- GROUP STATE
-- Group    System    Probed   AutoDisabled   State
B  sg1      node01    Y        N              ONLINE
B  sg1      node02    Y        N              ONLINE
B  sg1      node03    Y        N              ONLINE
B  sg1      node04    Y        N              ONLINE
B  sg2      node01    Y        N              ONLINE
B  sg2      node02    Y        N              ONLINE
B  sg2      node03    Y        N              ONLINE
B  sg2      node04    Y        N              ONLINE
B  sg3      node01    Y        N              ONLINE
B  sg3      node02    Y        N              OFFLINE
B  sg3      node03    Y        N              OFFLINE
B  sg3      node04    Y        N              OFFLINE
B  sg4      node01    Y        N              OFFLINE
B  sg4      node02    Y        N              ONLINE
B  sg4      node03    Y        N              OFFLINE
B  sg4      node04    Y        N              OFFLINE
```

5 After the upgrade is complete, start the VxVM volumes (for each disk group) and mount the VxFS file systems.

In this example, you have performed a phased upgrade of VCS. The service groups were down when you took them offline on node03 and node04, to the time VCS brought them online on node01 or node02.

Note: If you want to upgrade the application clusters that use CP server based fencing to version 6.1 and later, make sure that you first upgrade VCS or SFHA on the CP server systems to version 6.1 and later. And then, from 7.0.1 onwards, CP server supports only HTTPS based communication with its clients and IPM-based communication is no longer supported. CP server needs to be reconfigured if you upgrade the CP server with IPM-based CP server configured.

Performing an automated VCS upgrade using response files

This chapter includes the following topics:

Upgrading VCS using response files

Typically, you can use the response file that the installer generates after you perform VCS upgrade on one system to upgrade VCS on other systems.

To perform automated VCS upgrade

1. Make sure the systems where you want to upgrade VCS meet the upgrade requirements.
2. Make sure the pre-upgrade tasks are completed.
3. Copy the response file to the system where you want to upgrade VCS.

   See “Sample response file for full upgrade of VCS” on page 228.
4. Edit the values of the response file variables as necessary.

   See “Response file variables to upgrade VCS” on page 226.

5 Mount the product disc and navigate to the folder that contains the installation program.

6 Start the upgrade from the system to which you copied the response file. For example:

```
# ./installer -responsefile /tmp/response_file
```

Where /tmp/*response_file* is the response file's full path name.

Response file variables to upgrade VCS

Table 13-1 lists the response file variables that you can define to upgrade VCS.

Table 13-1 Response file variables specific to upgrading VCS

Variable	List or Scalar	Description
CFG{opt}{upgrade}	Scalar	Upgrades VCS RPMs. (Required)
CFG{accepteula}	Scalar	Specifies whether you agree with EULA.pdf on the media. (Required)
CFG{systems}	List	List of systems on which the product is to be upgraded. (Required)
CFG{defaultaccess}	Scalar (optional)	Defines if the user chooses to grant read access for VCS cluster information to everyone.
CFG{key}	Scalar (optional)	Stores the keyless key you want to register.
CFG{vcs_allowcomms}	Scalar	Indicates whether or not to start LLT and GAB when you set up a single-node cluster. The value can be 0 (do not start) or 1 (start). (Required)
CFG{opt}{keyfile}	Scalar	Defines the location of an ssh keyfile that is used to communicate with all remote systems. (Optional)

Table 13-1 Response file variables specific to upgrading VCS *(continued)*

Variable	List or Scalar	Description
CFG{opt}{pkgpath}	Scalar	Defines a location, typically an NFS mount, from which all remote systems can install product RPMs. The location must be accessible from all target systems. (Optional)
CFG{opt}{tmppath}	Scalar	Defines the location where a working directory is created to store temporary files and the RPMs that are needed during the install. The default location is /opt/VRTStmp. (Optional)
CFG{secusrgrps}	List	Defines the user groups which get read access to the cluster. (Optional)
CFG{opt}{logpath}	Scalar	Mentions the location where the log files are to be copied. The default location is /opt/VRTS/install/logs. **Note:** The installer copies the response files and summary files also to the specified *logpath* location. (Optional)
CFG{opt}{rsh}	Scalar	Defines that *rsh* must be used instead of ssh as the communication method between systems. (Optional)
CFG{opt}{online_upgrade}	Scalar	Set the value to 1 for online upgrades.

Table 13-1 Response file variables specific to upgrading VCS *(continued)*

Variable	List or Scalar	Description
CFG{edgeserver_host}	String	Use this parameter to configure the edge server. Enter **telemetry.veritas.com** to use the Veritas Cloud Receiver, which is a preconfigured, cloud-based edge server deployed by Veritas. Optional or required: required **Note:** An edge server is used to collect licensing and platform related information from InfoScale products as part of the Veritas Product Improvement Program. The information collected helps identify how customers deploy and use the product, and enables Veritas to manage customer licenses more efficiently.
CFG{edgeserver_port}	Scalar	Use this parameter to configure the port number of the edge server. Enter 443, which is the port number used by the Veritas Cloud Receiver. Optional or required: required **Note:** An edge server is used to collect licensing and platform related information from InfoScale products as part of the Veritas Product Improvement Program. The information collected helps identify how customers deploy and use the product, and enables Veritas to manage customer licenses more efficiently.

Sample response file for full upgrade of VCS

Review the response file variables and their definitions.

See “Response file variables to upgrade VCS” on page 226.

```
our %CFG;

$CFG{accepteula}=1;
$CFG{keys}{keyless}=[ "AVAILABILITY" ];
$CFG{opt}{gco}=1;
$CFG{opt}{redirect}=1;
$CFG{opt}{upgrade}=1;
$CFG{opt}{vr}=1;
$CFG{prod}="AVAILABILITY742";
$CFG{systems}=[ "sys01","sys02" ];
$CFG{vcs_allowcomms}=1;

1;
```

Section 5

Adding and removing cluster nodes

Adding a node to a single-node cluster

This chapter includes the following topics:

Adding a node to a single-node cluster

All nodes in the new cluster must run the same version of VCS. The example procedure refers to the existing single-node VCS node as Node A. The node that is to join Node A to form a multiple-node cluster is Node B.

Table 14-1 specifies the activities that you need to perform to add nodes to a single-node cluster.

Table 14-1 Tasks to add a node to a single-node cluster

Task	Reference
Set up Node B to be compatible with Node A.	See "Setting up a node to join the single-node cluster" on page 232.
■ Add Ethernet cards for private heartbeat network for Node B. ■ If necessary, add Ethernet cards for private heartbeat network for Node A. ■ Make the Ethernet cable connections between the two nodes.	See "Installing and configuring Ethernet cards for private network" on page 233.
Connect both nodes to shared storage.	See "Configuring the shared storage" on page 234.

Table 14-1 Tasks to add a node to a single-node cluster *(continued)*

Task	Reference
■ Bring up VCS on Node A. ■ Edit the configuration file.	See "Bringing up the existing node" on page 234.
If necessary, install VCS on Node B and add a license key. Make sure Node B is running the same version of VCS as the version on Node A.	See the Veritas InfoScale Installation Guide for installation instructions.
Edit the configuration files on Node B.	See "About the VCS configuration files" on page 273.
Start LLT and GAB on Node B.	See "Starting LLT and GAB" on page 235.
■ Start LLT and GAB on Node A. ■ Copy UUID from Node A to Node B. ■ Restart VCS on Node A. ■ Modify service groups for two nodes.	See "Reconfiguring VCS on the existing node" on page 236.
■ Start VCS on Node B. ■ Verify the two-node cluster.	See "Verifying configuration on both nodes" on page 237.

Setting up a node to join the single-node cluster

The new node to join the existing single node that runs VCS must run the same operating system.

To set up a node to join the single-node cluster

1 Do one of the following tasks:

- If VCS is not currently running on Node B, proceed to step 2.
- If the node you plan to add as Node B is currently part of an existing cluster, remove the node from the cluster. After you remove the node from the cluster, remove the VCS RPMs and configuration files.
  See "Removing a node from a VCS cluster" on page 257.
- If the node you plan to add as Node B is also currently a single VCS node, uninstall VCS.

- If you renamed the LLT and GAB startup files, remove them.

2 If necessary, install VxVM and VxFS.

See “Installing VxVM or VxFS if necessary” on page 233.

Installing VxVM or VxFS if necessary

If you have either VxVM or VxFS with the cluster option installed on the existing node, install the same version on the new node.

Refer to the appropriate documentation for VxVM and VxFS to verify the versions of the installed products. Make sure the same version runs on all nodes where you want to use shared storage.

Installing and configuring Ethernet cards for private network

Both nodes require Ethernet cards (NICs) that enable the private network. If both Node A and Node B have Ethernet cards installed, you can ignore this step.

For high availability, use two separate NICs on each node. The two NICs provide redundancy for heartbeating.

To install and configure Ethernet cards for private network

1 Shut down VCS on Node A.

```
# hastop -local
```

2 Shut down the node to get to the `OK` prompt:

```
# shutdown -r now
```

3 Install the Ethernet card on Node A.

If you want to use aggregated interface to set up private network, configure aggregated interface.

4 Install the Ethernet card on Node B.

If you want to use aggregated interface to set up private network, configure aggregated interface.

5 Configure the Ethernet card on both nodes.

6 Make the two Ethernet cable connections from Node A to Node B for the private networks.

7 Restart the nodes.

Configuring the shared storage

Make the connection to shared storage from Node B. Configure VxVM on Node B and reboot the node when you are prompted.

Bringing up the existing node

Bring up the node.

To bring up the node

1. Restart Node A.
2. Log in as superuser.
3. Make the VCS configuration writable.

   ```
   # haconf -makerw
   ```

4. Display the service groups currently configured.

   ```
   # hagrp -list
   ```

5. Freeze the service groups.

   ```
   # hagrp -freeze group -persistent
   ```

   Repeat this command for each service group in step 4.
6. Make the configuration read-only.

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

7. Stop VCS on Node A.

   ```
   # hastop -local -force
   ```

8 Edit the VCS system configuration file /etc/sysconfig/vcs as follows:

Change the line:

```
ONENODE=yes
```

To:

```
ONENODE=no
```

9 Enable the GAB and LLT startup files so they can be used.

```
# mv /opt/VRTSgab/gab.old /opt/VRTSgab/gab
# mv /opt/VRTSllt/llt.old /opt/VRTSllt/llt
```

Creating configuration files

Create the configuration files for your cluster.

To create the configuration files

1 Create the file /etc/llttab for a two-node cluster

See “Setting up /etc/llttab for a manual installation” on page 139.

2 Create the file /etc/llthosts that list both the nodes.

See “Setting up /etc/llthosts for a manual installation” on page 138.

3 Create the file /etc/gabtab.

See “Configuring GAB manually” on page 141.

Starting LLT and GAB

On the new node, start LLT and GAB.

To start LLT and GAB

1 Start LLT on Node B.

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

```
# systemctl start llt
```

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

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

2 Start GAB on Node B.

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

```
# systemctl start gab
```

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

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

Reconfiguring VCS on the existing node

Reconfigure VCS on the existing nodes.

To reconfigure VCS on existing nodes

1 On Node A, create the files /etc/llttab, /etc/llthosts, and /etc/gabtab. Use the files that are created on Node B as a guide, customizing the /etc/llttab for Node A.

2 Start LLT on Node A.

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

```
# systemctl start llt
```

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

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

3 Start GAB on Node A.

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

```
# systemctl start gab
```

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

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

4 Check the membership of the cluster.

```
# gabconfig -a
```

5. Copy the cluster UUID from the existing node to the new node:

   ```
   # /opt/VRTSvcs/bin/uuidconfig.pl -clus -copy -from_sys \
   node_name_in_running_cluster -to_sys new_sys1 ... new_sysn
   ```

   Where you are copying the cluster UUID from a node in the cluster (*node_name_in_running_cluster*) to systems from *new_sys1* through *new_sysn* that you want to join the cluster.

6. Start VCS on Node A.

   ```
   # hastart
   ```

7. Make the VCS configuration writable.

   ```
   # haconf -makerw
   ```

8. Add Node B to the cluster.

   ```
   # hasys -add sysB
   ```

9. Add Node B to the system list of each service group.

   - List the service groups.

     ```
     # hagrp -list
     ```

   - For each service group that is listed, add the node.

     ```
     # hagrp -modify group SystemList -add sysB 1
     ```

Verifying configuration on both nodes

Verify the configuration for the nodes.

To verify the nodes' configuration

1. On Node B, check the cluster membership.

   ```
   # gabconfig -a
   ```

2. Start the VCS on Node B.

   ```
   # hastart
   ```

3. Verify that VCS is up on both nodes.

   ```
   # hastatus
   ```

4. List the service groups.

   ```
   # hagrp -list
   ```

5. Unfreeze the service groups.

   ```
   # hagrp -unfreeze group -persistent
   ```

6. Save the new two-node configuration.

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

Chapter 15

Adding a node to a multi-node VCS cluster

This chapter includes the following topics:

Adding nodes using the VCS installer

The VCS installer performs the following tasks:

- Verifies that the node and the existing cluster meet communication requirements.
- Verifies the products and RPMs installed on the new node.
- Discovers the network interfaces on the new node and checks the interface settings.
- Creates the following files on the new node:

  `/etc/llttab`

  `/etc/VRTSvcs/conf/sysname`
- Updates the following configuration files and copies them on the new node:

  `/etc/llthosts`

  `/etc/gabtab`

  `/etc/VRTSvcs/conf/config/main.cf`
- Copies the following files from the existing cluster to the new node
  /etc/vxfenmode
  /etc/vxfendg
  /etc/vx/.uuids/clusuuid
  /etc/sysconfig/llt

/etc/sysconfig/gab

/etc/sysconfig/vxfen

- Configures disk-based or server-based fencing depending on the fencing mode in use on the existing cluster.

At the end of the process, the new node joins the VCS cluster.

Note: If you have configured server-based fencing on the existing cluster, make sure that the CP server does not contain entries for the new node. If the CP server already contains entries for the new node, remove these entries before adding the node to the cluster, otherwise the process may fail with an error.

To add the node to an existing VCS cluster using the VCS installer

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

2 Run the VCS installer with the -addnode option.

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

```
# ./installer  -addnode
```

The installer displays the copyright message and the location where it stores the temporary installation logs.

3 Enter the name of a node in the existing VCS cluster. The installer uses the node information to identify the existing cluster.

```
Enter the name of any one node of the InfoScale Availability cluster wher
would like to add one or more new nodes:  sys1
```

4 Review and confirm the cluster information.

5 Enter the name of the systems that you want to add as new nodes to the cluster.

```
Enter the system names separated by spaces
to add to the cluster: sys5
```

The installer checks the installed products and RPMs on the nodes and discovers the network interfaces.

6 Enter the name of the network interface that you want to configure as the first private heartbeat link.

Note: The LLT configuration for the new node must be the same as that of the existing cluster. If your existing cluster uses LLT over UDP, the installer asks questions related to LLT over UDP for the new node. If your existing cluster uses LLT over RDMA, the installer asks questions related to LLT over RDMA for the new node.

See “Configuring private heartbeat links” on page 53.

```
Enter the NIC for the first private heartbeat
link on sys5: [b,q,?] eth1
```

7 Enter the name of the network interface that you want to configure as the second private heartbeat link.

```
Enter the NIC for the second private heartbeat link
on sys5: [b,q,?] eth2
```

8 Depending on the number of LLT links configured in the existing cluster, configure additional private heartbeat links for the new node.

The installer verifies the network interface settings and displays the information.

9 Review and confirm the information.

10 If you have configured SMTP, SNMP, or the global cluster option in the existing cluster, you are prompted for the NIC information for the new node.

```
Enter the NIC for VCS to use on sys5: eth3
```

11 If you have enabled security on the cluster, the installer displays the following message:

```
Since the cluster is in secure mode, check the main.cf
whether you need to modify the usergroup that you would
like to grant read access. If needed, use the following
commands to modify:

haconf -makerw
hauser -addpriv <user group> GuestGroup
haconf -dump -makero
```

Manually adding a node to a cluster

The system you add to the cluster must meet the hardware and software requirements.

Table 15-1 specifies the tasks that are involved in adding a cluster. The example demonstrates how to add a node saturn to already existing nodes, sys1 and sys2.

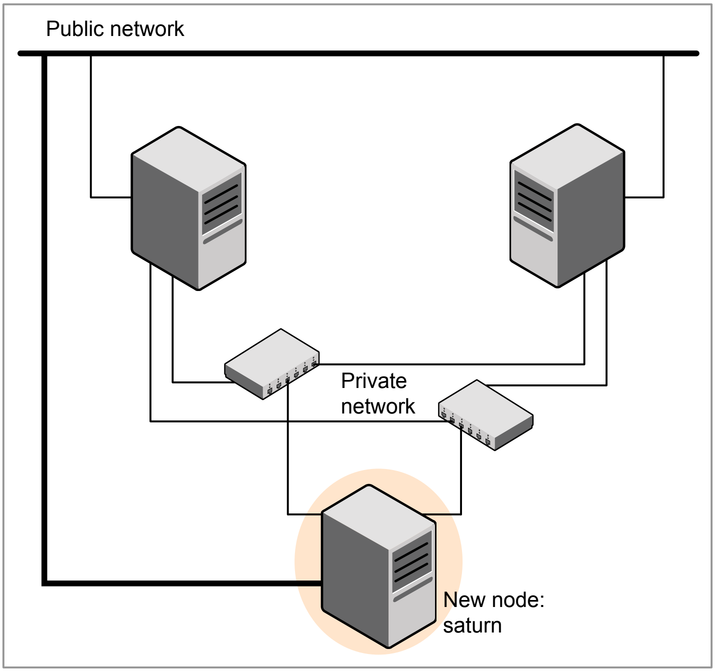
Table 15-1 Tasks that are involved in adding a node to a cluster

Task	Reference
Set up the hardware	See "Setting up the hardware" on page 242.
Configure LLT and GAB	See "Configuring LLT and GAB when adding a node to the cluster" on page 247.
Copy the UUID	See "Reconfiguring VCS on the existing node" on page 236.
Add the node to the existing cluster	See "Adding the node to the existing cluster" on page 253.
Start VCS and verify the cluster	See "Starting VCS and verifying the cluster" on page 254.

Setting up the hardware

Figure 15-1 shows that before you configure a new system on an existing cluster, you must physically add the system to the cluster.

Figure 15-1 Adding a node to a two-node cluster using two switches



To set up the hardware

1. Connect the VCS private Ethernet controllers.

   Perform the following tasks as necessary:

   - When you add nodes to a two-node cluster, use independent switches or hubs for the private network connections. You can only use crossover cables for a two-node cluster, so you might have to swap out the cable for a switch or hub.
   - If you already use independent hubs, connect the two Ethernet controllers on the new node to the independent hubs.

   Figure 15-1 illustrates a new node being added to an existing two-node cluster using two independent hubs.

2. Connect the system to the shared storage, if required.

Installing the VCS software manually when adding a node

Install the VCS 7.4.2 RPMs manually and add a license key.

Setting up the node to run in secure mode

You must follow this procedure only if you are adding a node to a cluster that is running in secure mode. If you are adding a node to a cluster that is not running in a secure mode, proceed with configuring LLT and GAB.

See "Configuring LLT and GAB when adding a node to the cluster" on page 247.

Table 15-2 uses the following information for the following command examples.

Table 15-2 The command examples definitions

Name	Fully-qualified host name (FQHN)	Function
sys5	sys5.nodes.example.com	The new node that you are adding to the cluster.

Configuring the authentication broker on node sys5

To configure the authentication broker on node sys5

1. Extract the embedded authentication files and copy them to temporary directory:

   ```
   # mkdir -p /var/VRTSvcs/vcsauth/bkup
   ```

   ```
   # cd /tmp; gunzip -c /opt/VRTSvcs/bin/VxAT.tar.gz | tar xvf -
   ```

2. Edit the setup file manually:

   ```
   # cat /etc/vx/.uuids/clusuuid 2>&1
   ```

   The output is a string denoting the UUID. This UUID (without { and }) is used as the ClusterName for the setup file.

   ```
   {UUID}
   ```

   ```
   # cat /tmp/eat_setup 2>&1
   ```

   The file content must resemble the following example:

   ```
   AcceptorMode=IP_ONLY

   BrokerExeName=vcsauthserver

   ClusterName=UUID

   DataDir=/var/VRTSvcs/vcsauth/data/VCSAUTHSERVER

   DestDir=/opt/VRTSvcs/bin/vcsauth/vcsauthserver

   FipsMode=0

   IPPort=14149

   RootBrokerName=vcsroot_uuid

   SetToRBPlusABorNot=0

   SetupPDRs=1

   SourceDir=/tmp/VxAT/version
   ```

3 Set up the embedded authentication file:

```
# cd /tmp/VxAT/version/bin/edition_number; \
./broker_setup.sh/tmp/eat_setup
```

```
# /opt/VRTSvcs/bin/vcsauth/vcsauthserver/bin/vssregctl -s -f
/var/VRTSvcs/vcsauth/data/VCSAUTHSERVER/root/.VRTSat/profile \
/VRTSatlocal.conf -b 'Security\Authentication \
\Authentication Broker' -k UpdatedDebugLogFileName \
-v /var/VRTSvcs/log/vcsauthserver.log -t string
```

4 Copy the broker credentials from one node in the cluster to sys5 by copying the entire `bkup` directory.

The `bkup` directory content resembles the following example:

```
# cd /var/VRTSvcs/vcsauth/bkup/
```

```
# ls
```

```
CMDSERVER  HAD  VCS_SERVICES  WAC
```

5 Import the VCS_SERVICES domain.

```
# /opt/VRTSvcs/bin/vcsauth/vcsauthserver/bin/atutil import -z \
/var/VRTSvcs/vcsauth/data/VCSAUTHSERVER -f /var/VRTSvcs/vcsauth/bkup \
/VCS_SERVICES -p password
```

6 Import the credentials for HAD, CMDSERVER, and WAC.

```
# /opt/VRTSvcs/bin/vcsauth/vcsauthserver/bin/atutil import -z \
/var/VRTSvcs/vcsauth/data/VCS_SERVICES -f /var/VRTSvcs/vcsauth/bkup \
/HAD -p password
```

7 Start the vcsauthserver process on sys5.

```
# /opt/VRTSvcs/bin/vcsauth/vcsauthserver/bin/vcsauthserver.sh
```

8. Perform the following tasks:

   ```
   # mkdir /var/VRTSvcs/vcsauth/data/CLIENT
   ```

   ```
   # mkdir /var/VRTSvcs/vcsauth/data/TRUST
   ```

   ```
   # export EAT_DATA_DIR='/var/VRTSvcs/vcsauth/data/TRUST'
   ```

   ```
   # /opt/VRTSvcs/bin/vcsauth/vcsauthserver/bin/vssat setuptrust -b \
   localhost:14149 -s high
   ```

9. Create the `/etc/VRTSvcs/conf/config/.secure` file:

   ```
   # touch /etc/VRTSvcs/conf/config/.secure
   ```

Configuring LLT and GAB when adding a node to the cluster

Create the LLT and GAB configuration files on the new node and update the files on the existing nodes.

To configure LLT when adding a node to the cluster

1. Create the file /etc/llthosts on the new node. You must also update it on each of the current nodes in the cluster.

   For example, suppose you add sys5 to a cluster consisting of sys1 and sys2:

   - If the file on one of the existing nodes resembles:

     ```
     0 sys1
     1 sys2
     ```

   - Update the file for all nodes, including the new one, resembling:

```
0 sys1
1 sys2
2 sys5
```

2 Create the file /etc/llttab on the new node, making sure that line beginning "`set-node`" specifies the new node.

The file /etc/llttab on an existing node can serve as a guide.

The following example describes a system where node sys2 is the new node on cluster ID number 2:

```
set-node sys2
set-cluster 2
link eth1 eth-00:04:23:AC:12:C4 - ether - -
link eth2 eth-00:04:23:AC:12:C5 - ether - -
```

3 Copy the following file from one of the nodes in the existing cluster to the new node:

/etc/sysconfig/llt

4 On the new system, run the command:

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

```
# systemctl start llt
```

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

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

In a setup that uses LLT over UDP, new nodes automatically join the existing cluster if the new nodes and all the existing nodes in the cluster are not separated by a router. However, if you use LLT over UDP6 link with IPv6 address and if the new node and the existing nodes are separated by a router, then do the following:

- Edit the `/etc/llttab` file on each node to reflect the link information about the new node.
- Specify the IPv6 address for UDP link of the new node to all existing nodes. Run the following command on each existing node for each UDP link:

  ```
  # /sbin/lltconfig -a set systemid device_tag address
  ```

To configure GAB when adding a node to the cluster

1 Create the file /etc/gabtab on the new system.

- If the /etc/gabtab file on the existing nodes resembles:

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

The file on the new node should be the same. Veritas recommends that you use the `-c -nN` option, where *N* is the total number of cluster nodes.

- If the /etc/gabtab file on the existing nodes resembles:

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

  The file on all nodes, including the new node, should change to reflect the change in the number of cluster nodes. For example, the new file on each node should resemble:

  ```
  /sbin/gabconfig -c -n3
  ```

  The `-n` flag indicates to VCS the number of nodes that must be ready to form a cluster before VCS starts.

2 Copy the following file from one of the nodes in the existing cluster to the new node:

/etc/sysconfig/gab

3 On the new node, to configure GAB run the command:

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

```
# systemctl start gab
```

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

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

To verify GAB

1 On the new node, run the command:

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

The output should indicate that port a membership shows all nodes including the new node. The output should resemble:

```
GAB Port Memberships
====================================
Port a gen a3640003 membership 012
```

2 Run the same command on the other nodes (sys1 and sys2) to verify that the port a membership includes the new node:

```
# /sbin/gabconfig -a
GAB Port Memberships
===================================
Port a gen a3640003 membership 012
Port h gen fd570002 membership 01
Port h gen fd570002    visible ; 2
```

Configuring I/O fencing on the new node

If the existing cluster is configured for I/O fencing, perform the following tasks on the new node:

- Prepare to configure I/O fencing on the new node.
  See "Preparing to configure I/O fencing on the new node" on page 250.
- If the existing cluster runs server-based fencing, configure server-based fencing on the new node.
  See "Configuring server-based fencing on the new node" on page 251.
  If the existing cluster runs disk-based fencing, you need not perform any additional step. Skip to the next task. After you copy the I/O fencing files and start I/O fencing, disk-based fencing automatically comes up.
- Copy the I/O fencing files from an existing node to the new node and start I/O fencing on the new node.
  See "Starting I/O fencing on the new node" on page 252.

If the existing cluster is not configured for I/O fencing, perform the procedure to add the new node to the existing cluster.

See "Adding the node to the existing cluster" on page 253.

Preparing to configure I/O fencing on the new node

Perform the following tasks before you configure and start I/O fencing on the new node.

To prepare to configure I/O fencing on the new node

1. Determine whether the existing cluster runs disk-based or server-based fencing mechanism. On one of the nodes in the existing cluster, run the following command:

   ```
   # vxfenadm -d
   ```

   If the fencing mode in the output is SCSI3, then the cluster uses disk-based fencing.

   If the fencing mode in the output is CUSTOMIZED, then the cluster uses server-based fencing.

2. In the following cases, install and configure Veritas Volume Manager (VxVM) on the new node.

   - The existing cluster uses disk-based fencing.
   - The existing cluster uses server-based fencing with at least one coordinator disk.

   You need not perform this step if the existing cluster uses server-based fencing with all coordination points as CP servers.

   See the *Veritas InfoScale Installation Guide* for installation instructions.

Configuring server-based fencing on the new node

This section describes the procedures to configure server-based fencing on a new node.

To configure server-based fencing on the new node

1. Log in to each CP server as the root user.
2. Update each CP server configuration with the new node information:

   ```
   # cpsadm -s cps1.example.com \
   -a add_node -c clus1 -h sys5 -n2

   Node 2 (sys5) successfully added
   ```

3. Verify that the new node is added to the CP server configuration:

   ```
   # cpsadm -s cps1.example.com -a list_nodes
   ```

   The new node must be listed in the output.

4. Copy the certificates to the new node from the peer nodes.

See "Generating the client key and certificates manually on the client nodes " on page 165.

Adding the new node to the vxfen service group

Perform the steps in the following procedure to add the new node to the vxfen service group.

To add the new node to the vxfen group using the CLI

1. On one of the nodes in the existing VCS cluster, set the cluster configuration to read-write mode:

   ```
   # haconf -makerw
   ```

2. Add the node sys5 to the existing vxfen group.

   ```
   # hagrp -modify vxfen SystemList -add sys5 2
   ```

3. Save the configuration by running the following command from any node in the VCS cluster:

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

Starting I/O fencing on the new node

Copy the I/O fencing files from an existing node to the new node and start I/O fencing on the new node. This task starts I/O fencing based on the fencing mechanism that is configured in the existing cluster.

To start I/O fencing on the new node

1. Copy the following I/O fencing configuration files from one of the nodes in the existing cluster to the new node:

   - /etc/vxfenmode
   - /etc/vxfendg—This file is required only for disk-based fencing.

- /etc/sysconfig/vxfen

2 Start I/O fencing on the new node.

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

```
# systemctl start vxfen
```

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

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

3 Run the GAB configuration command on the new node to verify that the port b membership is formed.

```
# gabconfig -a
```

Adding the node to the existing cluster

Perform the tasks on one of the existing nodes in the cluster.

To add the new node to the existing cluster

1 Enter the command:

```
# haconf -makerw
```

2 Add the new system to the cluster:

```
# hasys -add sys1
```

3 Copy the main.cf file from an existing node to your new node:

```
# rcp /etc/VRTSvcs/conf/config/main.cf \
  sys5:/etc/VRTSvcs/conf/config/
```

4 Check the VCS configuration file. No error message and a return value of zero indicates that the syntax is legal.

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

5 If necessary, modify any new system attributes.

6 Enter the command:

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

Starting VCS and verifying the cluster

Start VCS after adding the new node to the cluster and verify the cluster.

To start VCS and verify the cluster

1 Start VCS on the newly added system:

```
# hastart
```

2 Run the GAB configuration command on each node to verify that port a and port h include the new node in the membership:

```
# /sbin/gabconfig -a
  GAB Port Memberships
  ===================================
  Port a gen a3640003 membership 012
  Port h gen fd570002 membership 012
```

Adding a node using response files

Typically, you can use the response file that the installer generates on one system to add nodes to an existing cluster.

To add nodes using response files

1 Make sure the systems where you want to add nodes meet the requirements.

2 Make sure all the tasks required for preparing to add a node to an existing VCS cluster are completed.

3 Copy the response file to one of the systems where you want to add nodes.

See “Sample response file for adding a node to a VCS cluster” on page 255.

4 Edit the values of the response file variables as necessary.

See “Response file variables to add a node to a VCS cluster” on page 255.

5. Mount the product disc and navigate to the folder that contains the installation program.
6. Start adding nodes from the system to which you copied the response file. For example:

   ```
   # ./installer -responsefile /tmp/response_file
   ```

   Where /tmp/*response_file* is the response file's full path name.

   Depending on the fencing configuration in the existing cluster, the installer configures fencing on the new node. The installer then starts all the required processes and joins the new node to cluster. The installer indicates the location of the log file and summary file with details of the actions performed.

Response file variables to add a node to a VCS cluster

Table 15-3 lists the response file variables that you can define to add a node to an VCS cluster.

Table 15-3 Response file variables for adding a node to an VCS cluster

Variable	Description
$CFG{opt}{addnode}	Adds a node to an existing cluster. List or scalar: scalar Optional or required: required
$CFG{newnodes}	Specifies the new nodes to be added to the cluster. List or scalar: list Optional or required: required

Sample response file for adding a node to a VCS cluster

The following example shows a response file for adding a node to a VCS cluster.

```
our %CFG;

$CFG{clustersystems}=[ qw(sys1) ];
$CFG{newnodes}=[ qw(sys5) ];
$CFG{opt}{addnode}=1;
$CFG{opt}{configure}=1;
$CFG{opt}{vr}=1;

$CFG{prod}=" AVAILABILITY742";
```

```
$CFG{systems}=[ qw(sys1 sys5) ];
$CFG{vcs_allowcomms}=1;
$CFG{vcs_clusterid}=101;
$CFG{vcs_clustername}="clus1";
$CFG{vcs_lltlink1}{sys5}="eth1";
$CFG{vcs_lltlink2}{sys5}="eth2";

1;
```

Chapter 16

Removing a node from a VCS cluster

This chapter includes the following topics:

Removing a node from a VCS cluster

Table 16-1 specifies the tasks that are involved in removing a node from a cluster. In the example procedure, the cluster consists of nodes sys1, sys2, and sys5; node sys5 is to leave the cluster.

Table 16-1 Tasks that are involved in removing a node

Task	Reference
■ Back up the configuration file. ■ Check the status of the nodes and the service groups.	See “Verifying the status of nodes and service groups” on page 258.
■ Switch or remove any VCS service groups on the node departing the cluster. ■ Delete the node from VCS configuration.	See “Deleting the departing node from VCS configuration” on page 259.
Modify the llthosts(4) and gabtab(4) files to reflect the change.	See “Modifying configuration files on each remaining node” on page 262.
For a cluster that is running in a secure mode, remove the security credentials from the leaving node.	See “Removing security credentials from the leaving node ” on page 263.

Table 16-1 Tasks that are involved in removing a node *(continued)*

Task	Reference
On the node departing the cluster: - Modify startup scripts for LLT, GAB, and VCS to allow reboot of the node without affecting the cluster. - Unconfigure and unload the LLT and GAB utilities.	See "Unloading LLT and GAB and removing Veritas InfoScale Availability or Enterprise on the departing node" on page 263.

Verifying the status of nodes and service groups

Start by issuing the following commands from one of the nodes to remain in the cluster node sys1 or node sys2 in our example.

To verify the status of the nodes and the service groups

1. Make a backup copy of the current configuration file, main.cf.

   ```
   # cp -p /etc/VRTSvcs/conf/config/main.cf\
   /etc/VRTSvcs/conf/config/main.cf.goodcopy
   ```

2. Check the status of the systems and the service groups.

   ```
   # hastatus -summary

       -- SYSTEM STATE
       -- System      State         Frozen
       A  sys1    RUNNING        0
       A  sys2    RUNNING        0
       A  sys5      RUNNING        0

       -- GROUP STATE
       -- Group    System      Probed   AutoDisabled   State
       B  grp1     sys1      Y        N              ONLINE
       B  grp1     sys2      Y        N              OFFLINE
       B  grp2     sys1      Y        N              ONLINE
       B  grp3     sys2      Y        N              OFFLINE
       B  grp3     sys5    Y        N              ONLINE
       B  grp4     sys5    Y        N              ONLINE
   ```

   The example output from the `hastatus` command shows that nodes sys1, sys2, and sys5 are the nodes in the cluster. Also, service group grp3 is configured to run on node sys2 and node sys5, the departing node. Service group grp4 runs only on node sys5. Service groups grp1 and grp2 do not run on node sys5.

Deleting the departing node from VCS configuration

Before you remove a node from the cluster you need to identify the service groups that run on the node.

You then need to perform the following actions:

- Remove the service groups that other service groups depend on, or
- Switch the service groups to another node that other service groups depend on.

To remove or switch service groups from the departing node

1. Switch failover service groups from the departing node. You can switch grp3 from node sys5 to node sys2.

   ```
   # hagrp -switch grp3 -to sys2
   ```

2. Check for any dependencies involving any service groups that run on the departing node; for example, grp4 runs only on the departing node.

   ```
   # hagrp -dep
   ```

3. If the service group on the departing node requires other service groups—if it is a parent to service groups on other nodes—unlink the service groups.

   ```
   # haconf -makerw
   # hagrp -unlink grp4 grp1
   ```

   These commands enable you to edit the configuration and to remove the requirement grp4 has for grp1.

4. Stop VCS on the departing node:

   ```
   # hastop -sys sys5
   ```

5. Check the status again. The state of the departing node should be EXITED. Make sure that any service group that you want to fail over is online on other nodes.

   ```
   # hastatus -summary

       -- SYSTEM STATE
       -- System      State          Frozen
       A  sys1    RUNNING        0
       A  sys2      RUNNING        0
       A  sys5      EXITED         0

       -- GROUP STATE
       -- Group    System      Probed   AutoDisabled   State
       B  grp1     sys1      Y        N              ONLINE
       B  grp1     sys2      Y        N              OFFLINE
       B  grp2     sys1      Y        N              ONLINE
       B  grp3     sys2      Y        N              ONLINE
       B  grp3     sys5    Y        Y              OFFLINE
       B  grp4     sys5    Y        N              OFFLINE
   ```

6 Delete the departing node from the SystemList of service groups grp3 and grp4.

```
# haconf -makerw
# hagrp -modify grp3 SystemList -delete sys5
# hagrp -modify grp4 SystemList -delete sys5
```

Note: If sys5 was in the autostart list, then you need to manually add another system in the autostart list so that after reboot, the group comes online automatically.

7 For the service groups that run only on the departing node, delete the resources from the group before you delete the group.

```
# hagrp -resources grp4
    processx_grp4
    processy_grp4
# hares -delete processx_grp4
# hares -delete processy_grp4
```

8 Delete the service group that is configured to run on the departing node.

```
# hagrp -delete grp4
```

9 Check the status.

```
# hastatus -summary
    -- SYSTEM STATE
    -- System      State          Frozen
    A  sys1      RUNNING        0
    A  sys2      RUNNING        0
    A  sys5    EXITED         0

    -- GROUP STATE
    -- Group    System      Probed   AutoDisabled   State
    B  grp1     sys1      Y        N              ONLINE
    B  grp1     sys2      Y        N              OFFLINE
    B  grp2     sys1      Y        N              ONLINE
    B  grp3     sys2      Y        N              ONLINE
```

10 Delete the node from the cluster.

```
# hasys -delete sys5
```

11 Save the configuration, making it read only.

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

Modifying configuration files on each remaining node

Perform the following tasks on each of the remaining nodes of the cluster.

To modify the configuration files on a remaining node

1 If necessary, modify the /etc/gabtab file.

No change is required to this file if the `/sbin/gabconfig` command has only the argument `-c`. Veritas recommends using the `-nN` option, where *N* is the number of cluster systems.

If the command has the form `/sbin/gabconfig -c -nN`, where *N* is the number of cluster systems, make sure that *N* is not greater than the actual number of nodes in the cluster. When *N* is greater than the number of nodes, GAB does not automatically seed.

Veritas does not recommend the use of the `-c -x` option for `/sbin/gabconfig`.

2 Modify /etc/llthosts file on each remaining nodes to remove the entry of the departing node.

For example, change:

```
0 sys1
1 sys2
2 sys5
```

To:

```
0 sys1
1 sys2
```

Removing the node configuration from the CP server

After removing a node from a VCS cluster, perform the steps in the following procedure to remove that node's configuration from the CP server.

Note: The `cpsadm` command is used to perform the steps in this procedure. For detailed information about the `cpsadm` command, see the *Cluster Server Administrator's Guide*.

To remove the node configuration from the CP server

1 Log into the CP server as the root user.

2 View the list of VCS users on the CP server.

If the CP server is configured to use HTTPS-based communication, run the following command:

```
# cpsadm -s cp_server -a list_users
```

Where *cp_server* is the virtual IP/ virtual hostname of the CP server.

3 Remove the node entry from the CP server:

```
# cpsadm -s cp_server -a rm_node  -h sys5 -c clus1 -n 2
```

4 View the list of nodes on the CP server to ensure that the node entry was removed:

```
# cpsadm -s cp_server -a list_nodes
```

Removing security credentials from the leaving node

If the leaving node is part of a cluster that is running in a secure mode, you must remove the security credentials from node sys5. Perform the following steps.

To remove the security credentials

1 Stop the AT process.

```
# /opt/VRTSvcs/bin/vcsauth/vcsauthserver/bin/vcsauthserver.sh \
stop
```

2 Remove the credentials.

```
# rm -rf /var/VRTSvcs/vcsauth/data/
```

Unloading LLT and GAB and removing Veritas InfoScale Availability or Enterprise on the departing node

Perform the tasks on the node that is departing the cluster.

You can use script-based installer to uninstall Veritas InfoScale Availability or Enterprise on the departing node or perform the following manual steps.

If you have configured VCS as part of the InfoScale products, you may have to delete other dependent RPMs before you can delete all of the following ones.

To stop LLT and GAB and remove Veritas InfoScale Availability or Enterprise

1. If you had configured I/O fencing in enabled mode, then stop I/O fencing.

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

   ```
   # systemctl stop vxfen
   ```

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

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

2. Stop GAB and LLT:

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

   ```
   # systemctl stop gab
   # systemctl stop llt
   ```

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

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

3. To determine the RPMs to remove, enter:

   ```
   # rpm -qa |grep VRTS
   ```

4. To permanently remove the Availability or Enterprise RPMs from the system, use the `rpm -e` command. Start by removing the following RPMs, which may have been optionally installed, in the order shown:

   ```
   # rpm -e VRTSsfcpi
   # rpm -e VRTSvcswiz
   # rpm -e VRTSvbs
   # rpm -e VRTSsfmh
   # rpm -e VRTSvcsea
   # rpm -e VRTSvcsdr
   # rpm -e VRTSvcsag
   # rpm -e VRTScps
   # rpm -e VRTSvcs
   # rpm -e VRTSamf
   # rpm -e VRTSvxfen
   ```

```
# rpm -e VRTSgab
# rpm -e VRTSllt
# rpm -e VRTSspt
# rpm -e VRTSvlic
# rpm -e VRTSperl
```

5 Remove the LLT and GAB configuration files.

```
# rm /etc/llttab
# rm /etc/gabtab
# rm /etc/llthosts
```

Section 6

Installation reference

Appendix A

Services and ports

This appendix includes the following topics:

- About InfoScale Enterprise services and ports

About InfoScale Enterprise services and ports

If you have configured a firewall, ensure that the firewall settings allow access to the services and ports used by InfoScale Enterprise.

Table A-1 lists the services and ports used by InfoScale Enterprise .

Note: The port numbers that appear in bold are mandatory for configuring InfoScale Enterprise.

Table A-1 SFHA services and ports

Port Number	Protocol	Description	Process
4145	TCP/UDP	VVR Connection Server VCS Cluster Heartbeats	vxio
5634	HTTPS	Veritas Storage Foundation Messaging Service	xprtld
8199	TCP	Volume Replicator Administrative Service	vras
8989	TCP	VVR Resync Utility	vxreserver

Table A-1 SFHA services and ports *(continued)*

Port Number	Protocol	Description	Process
14141	TCP	Veritas High Availability Engine Veritas Cluster Manager (Java console) (ClusterManager.exe) VCS Agent driver (VCSAgDriver.exe)	had
14144	TCP/UDP	VCS Notification	Notifier
14149	TCP/UDP	VCS Authentication	vcsauthserver
14150	TCP	Veritas Command Server	CmdServer
14155	TCP/UDP	VCS Global Cluster Option (GCO)	wac
14156	TCP/UDP	VCS Steward for GCO	steward
443	TCP	Coordination Point Server	Vxcpserv
49152-65535	TCP/UDP	Volume Replicator Packets	User configurable ports created at kernel level by `vxio.sys` file

Configuration files

This appendix includes the following topics:

About the LLT and GAB configuration files

Low Latency Transport (LLT) and Group Membership and Atomic Broadcast (GAB) are VCS communication services. LLT requires /etc/llthosts and /etc/llttab files. GAB requires /etc/gabtab file.

Table B-1 lists the LLT configuration files and the information that these files contain.

Table B-1 LLT configuration files

File	Description
/etc/sysconfig/llt	This file stores the start and stop environment variables for LLT: ■ LLT_START—Defines the startup behavior for the LLT module after a system reboot. Valid values include: 1—Indicates that LLT is enabled to start up. 0—Indicates that LLT is disabled to start up. ■ LLT_STOP—Defines the shutdown behavior for the LLT module during a system shutdown. Valid values include: 1—Indicates that LLT is enabled to shut down. 0—Indicates that LLT is disabled to shut down. The installer sets the value of these variables to 1 at the end of VCS configuration. If you manually configured VCS, make sure you set the values of these environment variables to 1. Assign the buffer pool memory for RDMA operations: ■ LLT_BUFPOOL_MAXMEM—Maximum assigned memory that LLT can use for the LLT buffer pool. This buffer pool is used to allocate memory for RDMA operations and packet allocation, which are delivered to the LLT clients. The default value is calculated based on the total system memory, the minimum value is 1GB, and the maximum value is 10GB. You must specify the value in GB. Define the number of RDMA queue pairs per LLT link ■ LLT_RDMA_QPS — Maximum number of RDMA queue pairs per LLT link. You can change the value of this parameter in the `/etc/sysconfig/llt` file. The default value is 4. However, you can extend this value to maximum 8 queue pairs per LLT link. Enable or disable the adaptive window feature: ■ For performance reason, the adaptive window feature (LLT_ENABLE_AWINDOW) is enabled by default for 5 (cfs) and port 24(cvm). You can disable the adaptive window feature by manually changing the value of the LLT_ENABLE_AWINDOW parameter to zero. To enable adaptive window for ports other than 5 and 24, add the port numbers in LLT_AW_PORT_LIST separated by comma. For example: LLT_AW_PORT_LIST=: '"5,24,0,1,5,14"'. If you want to disable the adaptive window feature for any of the ports, remove that specific port from this parameter. Example: LLT_AW_PORT_LIST='"5"' Configure LLT over TCP ■ When you configure LLT over the Transmission Control Protocol (TCP) layer for clusters using wide-area networks and routers, you can use the LLT_TCP_CONNS parameter to control the number of TCP connections that can be established between peer nodes. The default value of this parameter is 8 connections. However, you can extend this value to maximum 64 connections.

Table B-1 LLT configuration files *(continued)*

File	Description

/etc/llthosts	The file `llthosts` is a database that contains one entry per system. This file links the LLT system ID (in the first column) with the LLT host name. This file must be identical on each node in the cluster. A mismatch of the contents of the file can cause indeterminate behavior in the cluster.
	For example, the file /etc/llthosts contains the entries that resemble:
	`0      sys1`
	`1      sys2`

/etc/llttab	The file `llttab` contains the information that is derived during installation and used by the utility `lltconfig(1M)`. After installation, this file lists the LLT network links that correspond to the specific system.
	For example, the file /etc/llttab contains the entries that resemble:
	`set-node sys1`
	`set-cluster 2`
	`link eth1 eth1 - ether - -`
	`link eth2 eth2 - ether - -`
	If you use aggregated interfaces, then the file contains the aggregated interface name instead of the eth-*MAC_address*.
	`set-node sys1`
	`set-cluster 2`
	`link eth1 eth-00:04:23:AC:12:C4 - ether - -`
	`link eth2 eth-00:04:23:AC:12:C5 - ether - -`
	The first line identifies the system. The second line identifies the cluster (that is, the cluster ID you entered during installation). The next two lines begin with the `link` command. These lines identify the two network cards that the LLT protocol uses.
	If you configured a low priority link under LLT, the file also includes a "link-lowpri" line.
	Refer to the `llttab(4)` manual page for details about how the LLT configuration may be modified. The manual page describes the ordering of the directives in the `llttab` file.

Table B-2 lists the GAB configuration files and the information that these files contain.

Table B-2 GAB configuration files

File	Description
/etc/sysconfig/gab	This file stores the start and stop environment variables for GAB: ■ GAB_START—Defines the startup behavior for the GAB module after a system reboot. Valid values include: 1—Indicates that GAB is enabled to start up. 0—Indicates that GAB is disabled to start up. ■ GAB_STOP—Defines the shutdown behavior for the GAB module during a system shutdown. Valid values include: 1—Indicates that GAB is enabled to shut down. 0—Indicates that GAB is disabled to shut down. The installer sets the value of these variables to 1 at the end of VCS configuration. If you manually configured VCS, make sure you set the values of these environment variables to 1.
/etc/gabtab	After you install VCS, the file /etc/gabtab contains a `gabconfig(1)` command that configures the GAB driver for use. The file /etc/gabtab contains a line that resembles: `/sbin/gabconfig -c -nN` The `-c` option configures the driver for use. The `-nN` specifies that the cluster is not formed until at least *N* nodes are ready to form the cluster. Veritas recommends that you set N to be the total number of nodes in the cluster. **Note:** Veritas does not recommend the use of the `-c -x` option for `/sbin/gabconfig`. Using `-c -x` can lead to a split-brain condition. Use the `-c` option for `/sbin/gabconfig` to avoid a split-brain condition.

About the AMF configuration files

Asynchronous Monitoring Framework (AMF) kernel driver provides asynchronous event notifications to the VCS agents that are enabled for intelligent resource monitoring.

Table B-3 lists the AMF configuration files.

Table B-3 AMF configuration files

File	Description
/etc/sysconfig/amf	This file stores the start and stop environment variables for AMF: ■ AMF_START—Defines the startup behavior for the AMF module after a system reboot or when AMF is attempted to start using the init script. Valid values include: 1—Indicates that AMF is enabled to start up. (default) 0—Indicates that AMF is disabled to start up. ■ AMF_STOP—Defines the shutdown behavior for the AMF module during a system shutdown or when AMF is attempted to stop using the init script. Valid values include: 1—Indicates that AMF is enabled to shut down. (default) 0—Indicates that AMF is disabled to shut down.
/etc/amftab	After you install VCS, the file `/etc/amftab` contains a `amfconfig`(1) command that configures the AMF driver for use. The AMF init script uses this `/etc/amftab` file to configure the AMF driver. The `/etc/amftab` file contains the following line by default: `/opt/VRTSamf/bin/amfconfig -c`

About the VCS configuration files

VCS configuration files include the following:

- main.cf
  The installer creates the VCS configuration file in the /etc/VRTSvcs/conf/config folder by default during the VCS configuration. The main.cf file contains the minimum information that defines the cluster and its nodes.
  See "Sample main.cf file for VCS clusters" on page 274.
  See "Sample main.cf file for global clusters" on page 276.

- types.cf
  The file types.cf, which is listed in the include statement in the main.cf file, defines the VCS bundled types for VCS resources. The file types.cf is also located in the folder /etc/VRTSvcs/conf/config.
  Additional files similar to types.cf may be present if agents have been added, such as OracleTypes.cf.

Note the following information about the VCS configuration file after installing and configuring VCS:

- The cluster definition includes the cluster information that you provided during the configuration. This definition includes the cluster name, cluster address, and the names of users and administrators of the cluster.

  Notice that the cluster has an attribute UserNames. The installer creates a user "admin" whose password is encrypted; the word "password" is the default password.
- If you set up the optional I/O fencing feature for VCS, then the UseFence = SCSI3 attribute is present.
- If you configured the cluster in secure mode, the main.cf includes "SecureClus = 1" cluster attribute.
- The installer creates the ClusterService service group if you configured the virtual IP, SMTP, SNMP, or global cluster options.

  The service group also has the following characteristics:
  - The group includes the IP and NIC resources.
  - The service group also includes the notifier resource configuration, which is based on your input to installer prompts about notification.
  - The installer also creates a resource dependency tree.
  - If you set up global clusters, the ClusterService service group contains an Application resource, wac (wide-area connector). This resource’s attributes contain definitions for controlling the cluster in a global cluster environment.

    Refer to the *Cluster Server Administrator's Guide* for information about managing VCS global clusters.

Refer to the *Cluster Server Administrator's Guide* to review the configuration concepts, and descriptions of main.cf and types.cf files for Linux systems.

Sample main.cf file for VCS clusters

The following sample main.cf file is for a cluster in secure mode.

```
include "types.cf"
include "OracleTypes.cf"
include "OracleASMTypes.cf"
include "Db2udbTypes.cf"
include "SybaseTypes.cf"


cluster vcs_cluster2 (
    UserNames = { admin = cDRpdxPmHpzS, smith = dKLhKJkHLh }
    ClusterAddress = "192.168.1.16"
    Administrators = { admin, smith }
    CounterInterval = 5
    SecureClus = 1
)

    system sys1 (
   )

    system sys2 (
   )

    group ClusterService (
       SystemList = { sys1 = 0, sys2 = 1 }
       UserStrGlobal = "LocalCluster@https://10.182.2.76:8443;"
       AutoStartList = { sys1, sys2 }
       OnlineRetryLimit = 3
       OnlineRetryInterval = 120
       )

    IP webip (
       Device = eth0
       Address = "192.168.1.16"
       NetMask = "255.255.240.0"
       )

NIC csgnic (
       Device = eth0
       NetworkHosts = { "192.168.1.17", "192.168.1.18" }
       )

NotifierMngr ntfr (
   SnmpConsoles = { "sys5" = Error, "sys4" = SevereError }
   SmtpServer = "smtp.example.com"
   SmtpRecipients =  { "ozzie@example.com" = Warning,
                  "harriet@example.com" = Error }
   )

webip requires csgnic
   ntfr requires csgnic

// resource dependency tree
//
//     group ClusterService
//     {
```

```
//     NotifierMngr ntfr
//         {
//         NIC csgnic
//         }
// }
```

Sample main.cf file for global clusters

If you installed VCS with the Global Cluster option, note that the ClusterService group also contains the Application resource, wac. The wac resource is required to control the cluster in a global cluster environment.

```
.
.
group ClusterService (
   SystemList = { sys1 = 0, sys2 = 1 }

   UserStrGlobal = "LocalCluster@https://10.182.2.78:8443;"

   AutoStartList = { sys1, sys2 }
   OnlineRetryLimit = 3
   OnlineRetryInterval = 120
   )

   Application wac (
      StartProgram = "/opt/VRTSvcs/bin/wacstart"
      StopProgram = "/opt/VRTSvcs/bin/wacstop"
      MonitorProcesses = { "/opt/VRTSvcs/bin/wac" }
      RestartLimit = 3
      )
.
.
```

In the following main.cf file example, bold text highlights global cluster specific entries.

```
include "types.cf"

cluster vcs03 (
    ClusterAddress = "10.182.13.50"
    SecureClus = 1
    )

system sysA (
    )
```

```
system sysB (
    )

system sysC (
    )

group ClusterService (
    SystemList = { sysA = 0, sysB = 1, sysC = 2 }
    AutoStartList = { sysA, sysB, sysC }
    OnlineRetryLimit = 3
    OnlineRetryInterval = 120
    )

Application wac (
    StartProgram = "/opt/VRTSvcs/bin/wacstart -secure"
    StopProgram = "/opt/VRTSvcs/bin/wacstop"
    MonitorProcesses = { "/opt/VRTSvcs/bin/wac -secure" }
    RestartLimit = 3
    )

IP gcoip (
    Device = eth0
    Address = "10.182.13.50"
    NetMask = "255.255.240.0"
    )

NIC csgnic (
    Device = eth0
    NetworkHosts = { "10.182.13.1" }
    )

NotifierMngr ntfr (
   SnmpConsoles = { sys4 = SevereError }
   SmtpServer = "smtp.example.com"
   SmtpRecipients =  { "ozzie@example.com" = SevereError }
   )

gcoip requires csgnic
ntfr requires csgnic
wac requires gcoip

// resource dependency tree
```

```
//
//     group ClusterService
//     {
//     NotifierMngr ntfr
//         {
//         NIC csgnic
//         }
//     Application wac
//         {
//         IP gcoip
//             {
//             NIC csgnic
//             }
//         }
//     }
```

About I/O fencing configuration files

Table B-4 lists the I/O fencing configuration files.

Table B-4 I/O fencing configuration files

File	Description
/etc/sysconfig/vxfen	This file stores the start and stop environment variables for I/O fencing: ■ VXFEN_START—Defines the startup behavior for the I/O fencing module after a system reboot. Valid values include: 1—Indicates that I/O fencing is enabled to start up. 0—Indicates that I/O fencing is disabled to start up. ■ VXFEN_STOP—Defines the shutdown behavior for the I/O fencing module during a system shutdown. Valid values include: 1—Indicates that I/O fencing is enabled to shut down. 0—Indicates that I/O fencing is disabled to shut down. The installer sets the value of these variables to 1 at the end of VCS configuration. If you manually configured VCS, you must make sure to set the values of these environment variables to 1.
/etc/vxfendg	This file includes the coordinator disk group information. This file is not applicable for server-based fencing and majority-based fencing.

Table B-4 I/O fencing configuration files *(continued)*

File	Description
/etc/vxfenmode	This file contains the following parameters: ■ vxfen_mode ■ scsi3—For disk-based fencing. ■ customized—For server-based fencing. ■ disabled—To run the I/O fencing driver but not do any fencing operations. ■ majority— For fencing without the use of coordination points. ■ vxfen_mechanism This parameter is applicable only for server-based fencing. Set the value as cps. ■ scsi3_disk_policy ■ dmp—Configure the vxfen module to use DMP devices The disk policy is dmp by default. If you use iSCSI devices, you must set the disk policy as dmp. **Note:** You must use the same SCSI-3 disk policy on all the nodes. ■ List of coordination points This list is required only for server-based fencing configuration. Coordination points in server-based fencing can include coordinator disks, CP servers, or both. If you use coordinator disks, you must create a coordinator disk group containing the individual coordinator disks. Refer to the sample file /etc/vxfen.d/vxfenmode_cps for more information on how to specify the coordination points and multiple IP addresses for each CP server. ■ single_cp This parameter is applicable for server-based fencing which uses a single highly available CP server as its coordination point. Also applicable for when you use a coordinator disk group with single disk. ■ autoseed_gab_timeout This parameter enables GAB automatic seeding of the cluster even when some cluster nodes are unavailable. This feature is applicable for I/O fencing in SCSI3 and customized mode. 0—Turns the GAB auto-seed feature on. Any value greater than 0 indicates the number of seconds that GAB must delay before it automatically seeds the cluster. -1—Turns the GAB auto-seed feature off. This setting is the default. ■ detect_false_pesb 0—Disables stale key detection. 1—Enables stale key detection to determine whether a preexisting split brain is a true condition or a false alarm. Default: 0 **Note:** This parameter is considered only when `vxfen_mode=customized`.

Table B-4 I/O fencing configuration files *(continued)*

File	Description

/etc/vxfentab	When I/O fencing starts, the vxfen startup script creates this /etc/vxfentab file on each node. The startup script uses the contents of the /etc/vxfendg and /etc/vxfenmode files. Any time a system is rebooted, the fencing driver reinitializes the vxfentab file with the current list of all the coordinator points.
	Note: The /etc/vxfentab file is a generated file; do not modify this file.
	For disk-based I/O fencing, the /etc/vxfentab file on each node contains a list of all paths to each coordinator disk along with its unique disk identifier. A space separates the path and the unique disk identifier. An example of the /etc/vxfentab file in a disk-based fencing configuration on one node resembles as follows:
	■ DMP disk:
	`/dev/vx/rdmp/sdx3 HITACHI%5F1724-100%20%20FAStT%5FDISKS%5F6`
	`00A0B8000215A5D000006804E795D0A3`
	`/dev/vx/rdmp/sdy3 HITACHI%5F1724-100%20%20FAStT%5FDISKS%5F6`
	`00A0B8000215A5D000006814E795D0B3`
	`/dev/vx/rdmp/sdz3 HITACHI%5F1724-100%20%20FAStT%5FDISKS%5F6`
	`00A0B8000215A5D000006824E795D0C3`
	For server-based fencing, the /etc/vxfentab file also includes the security settings information.
	For server-based fencing with single CP server, the /etc/vxfentab file also includes the single_cp settings information.
	This file is not applicable for majority-based fencing.

Sample configuration files for CP server

The `/etc/vxcps.conf` file determines the configuration of the coordination point server (CP server.)

See "Sample CP server configuration (/etc/vxcps.conf) file output" on page 286.

The following are example main.cf files for a CP server that is hosted on a single node, and a CP server that is hosted on an SFHA cluster.

- The main.cf file for a CP server that is hosted on a single node:
  See "Sample main.cf file for CP server hosted on a single node that runs VCS" on page 281.
- The main.cf file for a CP server that is hosted on an SFHA cluster:
  See "Sample main.cf file for CP server hosted on a two-node SFHA cluster" on page 283.

The example main.cf files use IPv4 addresses.

Sample main.cf file for CP server hosted on a single node that runs VCS

The following is an example of a single CP server node main.cf.

For this CP server single node main.cf, note the following values:

- Cluster name: cps1
- Node name: cps1

```
include "types.cf"
include "/opt/VRTScps/bin/Quorum/QuorumTypes.cf"

// cluster name:  cps1
// CP server: cps1

cluster cps1 (
     UserNames = { admin = bMNfMHmJNiNNlVNhMK, haris = fopKojNvpHouNn,
             "cps1.example.com@root@vx" = aj,
             "root@cps1.example.com" = hq }
     Administrators = { admin, haris,
             "cps1.example.com@root@vx",
             "root@cps1.example.com" }
     SecureClus = 1
     HacliUserLevel = COMMANDROOT
     )

system cps1 (
     )

group CPSSG (
      SystemList = { cps1 = 0 }
      AutoStartList = { cps1 }
      )

      IP cpsvip1 (
            Critical = 0
            Device @cps1 = eth0
            Address = "10.209.3.1"
            NetMask = "255.255.252.0"
            )
      IP cpsvip2 (
            Critical = 0
            Device @cps1 = eth1
            Address = "10.209.3.2"
            NetMask = "255.255.252.0"
            )

      NIC cpsnic1 (
           Critical = 0
           Device @cps1 = eth0
           PingOptimize = 0
           NetworkHosts @cps1 = { "10.209.3.10" }
           )

      NIC cpsnic2 (
           Critical = 0
           Device @cps1 = eth1
           PingOptimize = 0
           )

      Process vxcpserv (
           PathName = "/opt/VRTScps/bin/vxcpserv"
           ConfInterval = 30
           RestartLimit = 3
           )

      Quorum quorum (
            QuorumResources = { cpsvip1, cpsvip2 }
            )

 cpsvip1 requires cpsnic1
 cpsvip2 requires cpsnic2
 vxcpserv requires quorum


 // resource dependency tree
 //
 // group CPSSG
 // {
 // IP cpsvip1
 //     {
 //     NIC cpsnic1
 //     }
```

```
 // IP cpsvip2
 //     {
 //     NIC cpsnic2
 //     }
 // Process vxcpserv
 //     {
 //     Quorum quorum
 //     }
 // }
```

Sample main.cf file for CP server hosted on a two-node SFHA cluster

The following is an example of a main.cf, where the CP server is hosted on an SFHA cluster.

For this CP server hosted on an SFHA cluster main.cf, note the following values:

- Cluster name: cps1
- Nodes in the cluster: cps1, cps2

```
include "types.cf"
include "CFSTypes.cf"
include "CVMTypes.cf"
include "/opt/VRTScps/bin/Quorum/QuorumTypes.cf"


// cluster: cps1
// CP servers:
// cps1
// cps2

cluster cps1 (
     UserNames = { admin = ajkCjeJgkFkkIskEjh,
            "cps1.example.com@root@vx" = JK,
            "cps2.example.com@root@vx" = dl }
     Administrators = { admin, "cps1.example.com@root@vx",
            "cps2.example.com@root@vx" }
     SecureClus = 1
     )

system cps1 (
     )

system cps2 (
     )

group CPSSG (
      SystemList = { cps1 = 0, cps2 = 1 }
      AutoStartList = { cps1, cps2 } )

      DiskGroup cpsdg (
            DiskGroup = cps_dg
            )

      IP cpsvip1 (
            Critical = 0
            Device @cps1 = eth0
            Device @cps2 = eth0
            Address = "10.209.81.88"
            NetMask = "255.255.252.0"
            )

      IP cpsvip2 (
            Critical = 0
            Device @cps1 = eth1
            Device @cps2 = eth1
            Address = "10.209.81.89"
            NetMask = "255.255.252.0"
            )

      Mount cpsmount (
            MountPoint = "/etc/VRTScps/db"
            BlockDevice = "/dev/vx/dsk/cps_dg/cps_volume"
            FSType = vxfs
            FsckOpt = "-y"
            )

      NIC cpsnic1 (
           Critical = 0
           Device @cps1 = eth0
           Device @cps2 = eth0
           PingOptimize = 0
           NetworkHosts @cps1 = { "10.209.81.10 }
           )

      NIC cpsnic2 (
           Critical = 0
```

```
          Device @cps1 = eth1
          Device @cps2 = eth1
          PingOptimize = 0
          )

     Process vxcpserv (
           PathName = "/opt/VRTScps/bin/vxcpserv"
           )

     Quorum quorum (
           QuorumResources = { cpsvip1, cpsvip2 }
           )

     Volume cpsvol (
           Volume = cps_volume
           DiskGroup = cps_dg
           )

cpsmount requires cpsvol
cpsvip1 requires cpsnic1
cpsvip2 requires cpsnic2
cpsvol requires cpsdg
vxcpserv requires cpsmount
vxcpserv requires quorum


// resource dependency tree
//
// group CPSSG
// {
// IP cpsvip1
//     {
//     NIC cpsnic1
//     }
// IP cpsvip2
//     {
//     NIC cpsnic2
//     }
// Process vxcpserv
//     {
//     Quorum quorum
//     Mount cpsmount
//         {
```

```
//          Volume cpsvol
//              {
//              DiskGroup cpsdg
//              }
//          }
//      }
// }
```

Sample CP server configuration (/etc/vxcps.conf) file output

The following is an example of a coordination point server (CP server) configuration file `/etc/vxcps.conf` output.

```
##  The vxcps.conf file determines the
## configuration for Veritas CP Server.
cps_name=cps1
vip=[10.209.81.88]
vip=[10.209.81.89]:56789
vip_https=[10.209.81.88]:55443
vip_https=[10.209.81.89]
port=14250
port_https=443
security=1
db=/etc/VRTScps/db
ssl_conf_file=/etc/vxcps_ssl.properties
```

Tuning LLT variables for FSS environments

In an FSS environment, you can improve application IO performance and also improve memory consumption by tuning a LLT variable. The settings are different for LLT communication over RDMA supported hardware and LLT over Ethernet supported hardware.

Tuning LLT variables for RDMA links

On an LLT over RDMA network, the default RDMA receive-buffer size is 64K and the number of receive-buffers that can be allocated is 1000.

To tune the LLT variables for RDMA links

1. Append the `/etc/sysconfig/llt` file with the following value.

   ```
   LLT_MAXADVBUFS=4000
   ```

   When you increase the number of receive-buffers to a maximum value of 4000 the IO performance improves, provided the cluster nodes have sufficient memory.

2. If the typical IO size is lesser than the average IO size, you can avoid memory wastage by tuning down the receive buffer size to a minimum of 8K. By default, the receive buffer size is 64K.

   Append the /etc/sysconfig/llt file with the following value.

   ```
   LLT_ADVBUF_SIZE=8192
   ```

3. Append the /etc/llttab file with the following values.

   ```
   set-flow highwater:10000
   set-flow lowwater:8000
   set-flow window:5000
   set-flow rporthighwater:10000
   set-flow rportlowwater:8000
   ```

4. Restart the SFCFS or SFCFSHA stack for changes to take effect.

To verify the LLT tunable values

◆ Check the LLT configuration.

```
# lltstat -c
```

```
LLT configuration information:
    node: 0
.
.
    max advertised buffers: 4000
    advertised buffer size: 8192
```

Tuning LLT variables for Ethernet links

Perform the steps below to tune LLT variables for Ethernet links.

1. Append the /etc/llttab file with the following values.

   ```
   set-flow highwater:10000
   set-flow lowwater:8000
   set-flow window:5000
   set-flow rporthighwater:10000
   set-flow rportlowwater:8000
   ```

2. Verify whether the LLT file is updated.

   ```
   # lltconfig -F query
   ```

3. Restart the SFCFS or SFCFSHA stack for changes to take effect.

Configuring parallel ports for I/O shipping

Cluster Volume Manager (CVM) uses a single port, port y, for shipping I/O from node(s). This provides only one channel for data transmission.

In an FSS environment, you can now tune LLT variables to use multiple parallel ports for I/O shipping. Using the LLT_NMULTIPORTS variable, you can set the number of ports that can provide multiple channels for the data transmission. Default value for this variable is 4. However, you can set up to 8 parallel ports for I/O shipping. These ports are internal to LLT and are not visible at GAB or CVM layer.

The data received from the CVM on port y is distributed internally among the ports, providing improved performance for I/O shipping.

Similarly, using the LLT_RDLV_MTFACTOR variable, you can change the number of threads per I/O shipping port. Default value is 2 that can be incremented up to 16.

To tune the LLT variables for I/O shipping

Append the `/etc/sysconfig/llt` file with the following value:

- To enable parallel ports for I/O shipping:
  LLT_NMULTIPORTS = *number of ports*
  If you set the value to 0 or 1, CVM uses only one port for I/O shipping.
- To set number of threads per I/O shipping port:
  LLT_RDLV_MTFACTOR = *number of threads*

If LLT_NMULTIPORTS = 0, the total number of threads for I/O shipping on the single port is 2. However, for performance reason, it is recommended that you set the number of threads to 8 or 16.

- To get the total number of ports, including parallel ports, registered with LLT:

  Use the `lltstat -p` command

  Note that this command does not provide information about which ports used as parallel ports.
- To get the number of ports used as parallel port for I/O shipping:

  Use the `lltstat -c` command

Configuring LLT over UDP

This appendix includes the following topics:

Using the UDP layer for LLT

VCS provides the option of using LLT over the UDP (User Datagram Protocol) layer for clusters using wide-area networks and routers. UDP makes LLT packets routable and thus able to span longer distances more economically.

When to use LLT over UDP

Use LLT over UDP in the following situations:

- LLT must be used over WANs
- When hardware, such as blade servers, do not support LLT over Ethernet

LLT over UDP is slower than LLT over Ethernet. Use LLT over UDP only when the hardware configuration makes it necessary.

Manually configuring LLT over UDP using IPv4

The following checklist is to configure LLT over UDP:

- Make sure that the LLT private links are on separate subnets. Set the broadcast address in /etc/llttab explicitly depending on the subnet for each link.
  See "Broadcast address in the /etc/llttab file" on page 291.
- Make sure that each NIC has an IP address that is configured before configuring LLT.
- Make sure the IP addresses in the /etc/llttab files are consistent with the IP addresses of the network interfaces.
- Make sure that each link has a unique not well-known UDP port.
  See "Selecting UDP ports" on page 293.
- Set the broadcast address correctly for direct-attached (non-routed) links.
  See "Sample configuration: direct-attached links" on page 294.
- For the links that cross an IP router, disable broadcast features and specify the IP address of each link manually in the /etc/llttab file.
  See "Sample configuration: links crossing IP routers" on page 296.

Broadcast address in the /etc/llttab file

The broadcast address is set explicitly for each link in the following example.

- Display the content of the /etc/llttab file on the first node sys1:

  ```
  sys1 # cat /etc/llttab

  set-node sys1
  set-cluster 1
  link link1 udp - udp  50000  -  192.168.9.1 192.168.9.255
  link link2 udp - udp  50001  -  192.168.10.1 192.168.10.255
  ```

  Verify the subnet mask using the ifconfig command to ensure that the two links are on separate subnets.
- Display the content of the /etc/llttab file on the second node sys2:

  ```
  sys2 # cat /etc/llttab

  set-node sys2
  set-cluster 1
  link link1 udp - udp  50000  -  192.168.9.2 192.168.9.255
  link link2 udp - udp  50001  -  192.168.10.2 192.168.10.255
  ```

  Verify the subnet mask using the ifconfig command to ensure that the two links are on separate subnets.

The link command in the /etc/llttab file

Review the link command information in this section for the /etc/llttab file. See the following information for sample configurations:

- See "Sample configuration: direct-attached links" on page 294.
- See "Sample configuration: links crossing IP routers" on page 296.

Table C-1 describes the fields of the link command that are shown in the /etc/llttab file examples. Note that some of the fields differ from the command for standard LLT links.

Table C-1 Field description for link command in /etc/llttab

Field	Description
tag-name	A unique string that is used as a tag by LLT; for example link1, link2,....
device	The device path of the UDP protocol; for example udp. A place holder string. On other unix platforms like Solaris or HP, this entry points to a device file (for example, /dev/udp). Linux does not have devices for protocols. So this field is ignored.
node-range	Nodes using the link. "-" indicates all cluster nodes are to be configured for this link.
link-type	Type of link; must be "udp" for LLT over UDP.
udp-port	Unique UDP port in the range of 49152-65535 for the link. See "Selecting UDP ports" on page 293.
MTU	"-" is the default, which has a value of 8192. The value may be increased or decreased depending on the configuration. Use the `lltstat -l` command to display the current value.
IP address	IP address of the link on the local node.
bcast-address	■ For clusters with enabled broadcasts, specify the value of the subnet broadcast address. ■ "-" is the default for clusters spanning routers.

The set-addr command in the /etc/llttab file

The `set-addr` command in the /etc/llttab file is required when the broadcast feature of LLT is disabled, such as when LLT must cross IP routers.

See "Sample configuration: links crossing IP routers" on page 296.

Table C-2 describes the fields of the set-addr command.

Table C-2 Field description for set-addr command in /etc/llttab

Field	Description
node-id	The node ID of the peer node; for example, 0.
link tag-name	The string that LLT uses to identify the link; for example link1, link2,....
address	IP address assigned to the link for the peer node.

Selecting UDP ports

When you select a UDP port, select an available 16-bit integer from the range that follows:

- Use available ports in the private range 49152 to 65535
- Do not use the following ports:
  - Ports from the range of well-known ports, 0 to 1023
  - Ports from the range of registered ports, 1024 to 49151

To check which ports are defined as defaults for a node, examine the file /etc/services. You should also use the `netstat` command to list the UDP ports currently in use. For example:

```
# netstat -au | more
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address Foreign Address         State
udp        0      0 *:32768                *:*
udp        0      0 *:956                  *:*
udp        0      0 *:tftp                 *:*
udp        0      0 *:sunrpc               *:*
udp        0      0 *:ipp                  *:*
```

Look in the UDP section of the output; the UDP ports that are listed under Local Address are already in use. If a port is listed in the /etc/services file, its associated name is displayed rather than the port number in the output.

Configuring the netmask for LLT

For nodes on different subnets, set the netmask so that the nodes can access the subnets in use. Run the following command and answer the prompt to set the netmask:

```
# ifconfig interface_name netmask netmask
```

For example:

- For the first network interface on the node sys1:

  ```
  IP address=192.168.9.1, Broadcast address=192.168.9.255,
  Netmask=255.255.255.0
  ```

  For the first network interface on the node sys2:

  ```
  IP address=192.168.9.2, Broadcast address=192.168.9.255,
  Netmask=255.255.255.0
  ```

- For the second network interface on the node sys1:

  ```
  IP address=192.168.10.1, Broadcast address=192.168.10.255,
  Netmask=255.255.255.0
  ```

  For the second network interface on the node sys2:

  ```
  IP address=192.168.10.2, Broadcast address=192.168.10.255,
  Netmask=255.255.255.0
  ```

Configuring the broadcast address for LLT

For nodes on different subnets, set the broadcast address in /etc/llttab depending on the subnet that the links are on.

An example of a typical /etc/llttab file when nodes are on different subnets. Note the explicitly set broadcast address for each link.

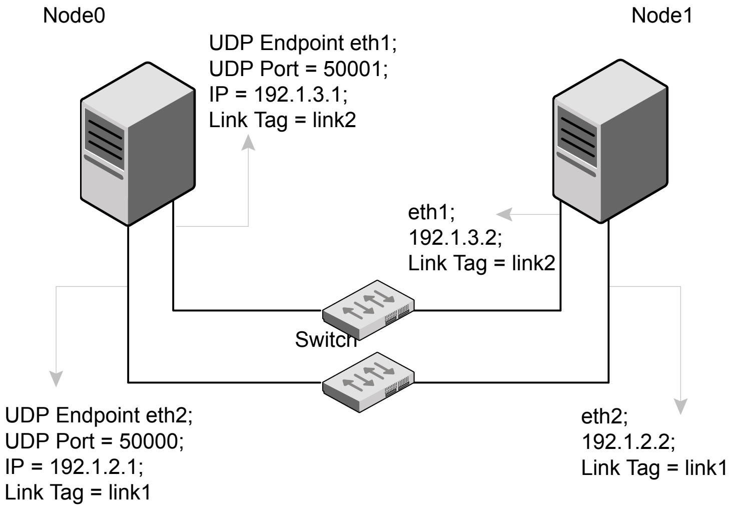
```
# cat /etc/llttab
set-node nodexyz
set-cluster 100

link link1 udp - udp 50000 - 192.168.30.1 192.168.30.255
link link2 udp - udp 50001 - 192.168.31.1 192.168.31.255
```

Sample configuration: direct-attached links

Figure C-1 depicts a typical configuration of direct-attached links employing LLT over UDP.

Figure C-1 A typical configuration of direct-attached links that use LLT over UDP



The configuration that the /etc/llttab file for Node 0 represents has directly attached crossover links. It might also have the links that are connected through a hub or switch. These links do not cross routers.

LLT sends broadcast requests to peer nodes to discover their addresses. So the addresses of peer nodes do not need to be specified in the /etc/llttab file using the `set-addr` command. For direct attached links, you do need to set the broadcast address of the links in the /etc/llttab file. Verify that the IP addresses and broadcast addresses are set correctly by using the `ifconfig -a` command.

```
set-node Node0
set-cluster 1
#configure Links
#link tag-name device node-range link-type udp port MTU \
IP-address bcast-address
link link1 udp - udp 50000 - 192.1.2.1 192.1.2.255
link link2 udp - udp 50001 - 192.1.3.1 192.1.3.255
```

The file for Node 1 resembles:

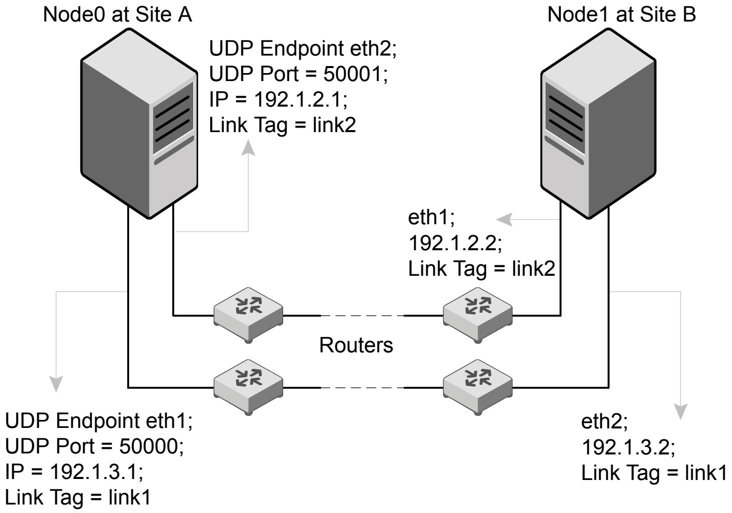
```
set-node Node1
set-cluster 1
#configure Links
#link tag-name device node-range link-type udp port MTU \
IP-address bcast-address
```

```
link link1 udp - udp 50000 - 192.1.2.2 192.1.2.255
link link2 udp - udp 50001 - 192.1.3.2 192.1.3.255
```

Sample configuration: links crossing IP routers

Figure C-2 depicts a typical configuration of links crossing an IP router employing LLT over UDP. The illustration shows two nodes of a four-node cluster.

Figure C-2 A typical configuration of links crossing an IP router



The configuration that the following `/etc/llttab` file represents for Node 1 has links crossing IP routers. Notice that IP addresses are shown for each link on each peer node. In this configuration broadcasts are disabled. Hence, the broadcast address does not need to be set in the `link` command of the `/etc/llttab` file.

```
set-node Node1
set-cluster 1

link link1 udp - udp 50000 - 192.1.3.1 -
link link2 udp - udp 50001 - 192.1.4.1 -

#set address of each link for all peer nodes in the cluster
#format: set-addr node-id link tag-name address
set-addr        0 link1 192.1.1.1
set-addr        0 link2 192.1.2.1
set-addr        2 link1 192.1.5.2
set-addr        2 link2 192.1.6.2
```

```
set-addr        3 link1 192.1.7.3
set-addr        3 link2 192.1.8.3

#disable LLT broadcasts
set-bcasthb     0
set-arp         0
```

The /etc/llttab file on Node 0 resembles:

```
set-node Node0
set-cluster 1

link link1 udp - udp 50000 - 192.1.1.1 -
link link2 udp - udp 50001 - 192.1.2.1 -

#set address of each link for all peer nodes in the cluster
#format: set-addr node-id link tag-name address
set-addr        1 link1 192.1.3.1
set-addr        1 link2 192.1.4.1
set-addr        2 link1 192.1.5.2
set-addr        2 link2 192.1.6.2
set-addr        3 link1 192.1.7.3
set-addr        3 link2 192.1.8.3

#disable LLT broadcasts
set-bcasthb     0
set-arp         0
```

Manually configuring LLT over UDP using IPv6

The following checklist is to configure LLT over UDP:

- For UDP6, the multicast address is set to "-".
- Make sure that each NIC has an IPv6 address that is configured before configuring LLT.
- Make sure the IPv6 addresses in the /etc/llttab files are consistent with the IPv6 addresses of the network interfaces.
- Make sure that each link has a unique not well-known UDP port.
  See “Selecting UDP ports” on page 299.
- For the links that cross an IP router, disable multicast features and specify the IPv6 address of each link manually in the /etc/llttab file.
  See “Sample configuration: links crossing IP routers” on page 301.

The link command in the /etc/llttab file

Review the link command information in this section for the /etc/llttab file. See the following information for sample configurations:

- See "Sample configuration: direct-attached links" on page 299.
- See "Sample configuration: links crossing IP routers" on page 301.

Note that some of the fields in Table C-3 differ from the command for standard LLT links.

Table C-3 describes the fields of the link command that are shown in the /etc/llttab file examples.

Table C-3 Field description for link command in /etc/llttab

Field	Description
tag-name	A unique string that is used as a tag by LLT; for example link1, link2,....
device	The device name of the UDP protocol; for example udp6.
node-range	Nodes using the link. "-" indicates all cluster nodes are to be configured for this link.
link-type	Type of link; must be "udp6" for LLT over UDP.
udp-port	Unique UDP port in the range of 49152-65535 for the link. See "Selecting UDP ports" on page 299.
MTU	"-" is the default, which has a value of 8192. The value may be increased or decreased depending on the configuration. Use the `lltstat -l` command to display the current value.
IPv6 address	IPv6 address of the link on the local node.
mcast-address	"-" is the default for clusters spanning routers.

The set-addr command in the /etc/llttab file

The `set-addr` command in the /etc/llttab file is required when the multicast feature of LLT is disabled, such as when LLT must cross IP routers.

See "Sample configuration: links crossing IP routers" on page 301.

Table C-4 describes the fields of the set-addr command.

Table C-4 Field description for set-addr command in /etc/llttab

Field	Description
node-id	The ID of the peer node; for example, 0.
link tag-name	The string that LLT uses to identify the link; for example link1, link2,....
address	IPv6 address assigned to the link for the peer node.

Selecting UDP ports

When you select a UDP port, select an available 16-bit integer from the range that follows:

- Use available ports in the private range 49152 to 65535
- Do not use the following ports:
  - Ports from the range of well-known ports, 0 to 1023
  - Ports from the range of registered ports, 1024 to 49151

To check which ports are defined as defaults for a node, examine the file /etc/services. You should also use the `netstat` command to list the UDP ports currently in use. For example:

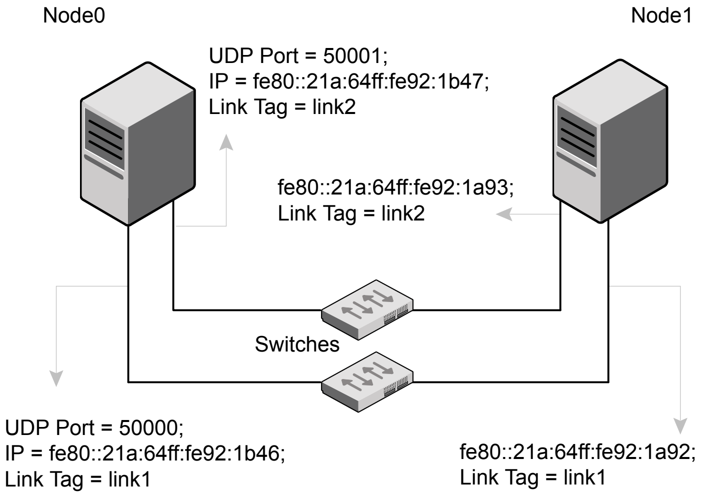
```
# netstat -au | more
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address Foreign Address         State
udp        0      0 *:32768                *:*
udp        0      0 *:956                  *:*
udp        0      0 *:tftp                 *:*
udp        0      0 *:sunrpc               *:*
udp        0      0 *:ipp                  *:*
```

Look in the UDP section of the output; the UDP ports that are listed under Local Address are already in use. If a port is listed in the /etc/services file, its associated name is displayed rather than the port number in the output.

Sample configuration: direct-attached links

Figure C-3 depicts a typical configuration of direct-attached links employing LLT over UDP.

Figure C-3 A typical configuration of direct-attached links that use LLT over UDP



The configuration that the /etc/llttab file for Node 0 represents has directly attached crossover links. It might also have the links that are connected through a hub or switch. These links do not cross routers.

LLT uses IPv6 multicast requests for peer node address discovery. So the addresses of peer nodes do not need to be specified in the /etc/llttab file using the `set-addr` command. Use the `ifconfig -a` command to verify that the IPv6 address is set correctly.

```
set-node Node0
set-cluster 1
#configure Links
#link tag-name device node-range link-type udp port MTU \
IP-address mcast-address
link link1 udp6 - udp6 50000 - fe80::21a:64ff:fe92:1b46 -
link link1 udp6 - udp6 50001 - fe80::21a:64ff:fe92:1b47 -
```

The file for Node 1 resembles:

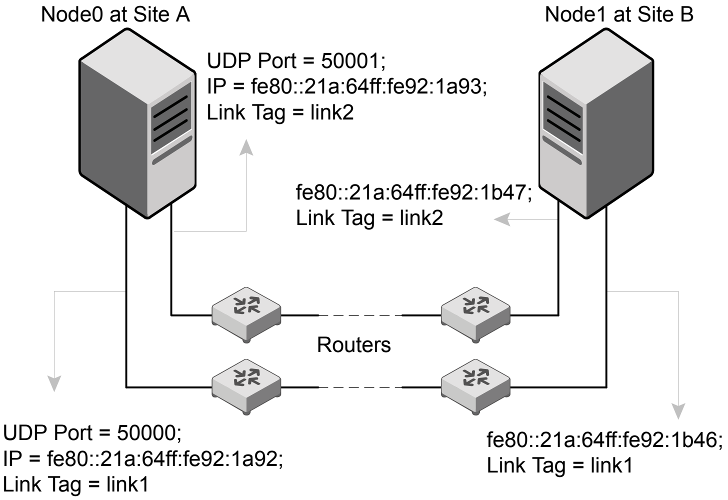
```
set-node Node1
set-cluster 1
#configure Links
#link tag-name device node-range link-type udp port MTU \
IP-address mcast-address
```

```
link link1 udp6 - udp6 50000 - fe80::21a:64ff:fe92:1a92 -
link link1 udp6 - udp6 50001 - fe80::21a:64ff:fe92:1a93 -
```

Sample configuration: links crossing IP routers

Figure C-4 depicts a typical configuration of links crossing an IP router employing LLT over UDP. The illustration shows two nodes of a four-node cluster.

Figure C-4 A typical configuration of links crossing an IP router



The configuration that the following `/etc/llttab` file represents for Node 1 has links crossing IP routers. Notice that IPv6 addresses are shown for each link on each peer node. In this configuration multicasts are disabled.

```
set-node Node1
set-cluster 1

link link1 udp6 - udp6 50000 - fe80::21a:64ff:fe92:1a92 -
link link1 udp6 - udp6 50001 - fe80::21a:64ff:fe92:1a93 -

#set address of each link for all peer nodes in the cluster
#format: set-addr node-id link tag-name address
set-addr 0 link1 fe80::21a:64ff:fe92:1b46
set-addr 0 link2 fe80::21a:64ff:fe92:1b47
set-addr 2 link1 fe80::21a:64ff:fe92:1d70
set-addr 2 link2 fe80::21a:64ff:fe92:1d71
set-addr 3 link1 fe80::209:6bff:fe1b:1c94
```

```
set-addr 3 link2 fe80::209:6bff:fe1b:1c95

#disable LLT multicasts
set-bcasthb     0
set-arp         0
```

The /etc/llttab file on Node 0 resembles:

```
set-node Node0
set-cluster 1

link link1 udp6 - udp6 50000 - fe80::21a:64ff:fe92:1b46 -
link link2 udp6 - udp6 50001 - fe80::21a:64ff:fe92:1b47 -

#set address of each link for all peer nodes in the cluster
#format: set-addr node-id link tag-name address
set-addr 1 link1 fe80::21a:64ff:fe92:1a92
set-addr 1 link2 fe80::21a:64ff:fe92:1a93
set-addr 2 link1 fe80::21a:64ff:fe92:1d70
set-addr 2 link2 fe80::21a:64ff:fe92:1d71
set-addr 3 link1 fe80::209:6bff:fe1b:1c94
set-addr 3 link2 fe80::209:6bff:fe1b:1c95

#disable LLT multicasts
set-bcasthb     0
set-arp         0
```

LLT over UDP sample /etc/llttab

The following is a sample of LLT over UDP in the etc/llttab file.

```
set-node sys1
set-cluster clus1
link eth1 udp - udp 50000 - 192.168.10.1 -
link eth2 udp - udp 50001 - 192.168.11.1 -
link-lowpri eth0 udp - udp 50004 - 10.200.58.205 -
set-addr 1 eth1 192.168.10.2
set-addr 1 eth2 192.168.11.2
set-addr 1 eth0 10.200.58.206
set-bcasthb 0
set-arp 0
```

About configuring LLT over UDP multiport

LLT uses UDP sockets for communication among the cluster nodes and creates one UDP socket per LLT link. In a Flexible Storage Sharing (FSS) environment, data can be read from and written to remote disks. In such a case, one socket per LLT link may not be enough for large read-write operations. More sockets are needed to achieve parallelism and throughput to meet the needs of the high data-generating applications.

Configuring LLT over UDP multiport enables you to create additional sockets per link. These sockets are reserved only for I/O shipping.

Note: For the multiport feature to work, LLT requires at least six consecutive network port numbers to be configured.

Manually configuring LLT over UDP multiport

Perform the following steps to configure LLT over UDP multiport.

Preparing for configuration

1. Set the maximum transmission unit (MTU) to the highest value (9000) supported by the NICs when the LLT high priority links are configured over UDP.

   Ensure that the network path MTU is also set to 9000.

   To change the MTU size permanently under Linux:

   a. Edit the `/etc/sysconfig/network-scripts/ifcfg-eth0` file

   ```
   # vi /etc/sysconfig/network-scripts/ifcfg-eth0
   ```

   b. Add MTU settings at the end of the file:

   MTU=9000

   c. Save and close the file and restart networking

   ```
   # service network restart
   ```

2. Enable the LLT ports in firewall. See “Enabling LLT ports in firewall” on page 305.

Configuring LLT over UDP Multiport

1. Use the following shell script to increase the size of network buffers which consequently increases the send and receive TCP/IP buffers. This script also tunes the Rx /Tx queue size to max and enables the receive side scaling (RSS) functionality of the NIC.

   ```
   #---------------------------------------
   set -x

   for card in `cat /etc/llttab | grep -v "lowpri" | grep -w "link" |
                    awk '{print $2}'`;
   do
       echo -e "Changeing buffers of $card"
       ethtool -G $card rx 4096
       ethtool -G $card rx-jumbo 4096
       ethtool -G $card tx 4096
       ethtool -N $card rx-flow-hash udp4 sdfn ethtool -N
                          $card rx-flow-hash tcp4 sdfn
       sysctl -w net.ipv4.conf.${card}.arp_ignore=1
   done

   sysctl -w net.core.rmem_max=1600000000
   sysctl -w net.core.wmem_max=1600000000
   sysctl -w net.core.netdev_max_backlog=250000
   sysctl -w net.core.rmem_default=4194304
   sysctl -w net.core.wmem_default=4194304
   sysctl -w net.core.optmem_max=4194304
   sysctl -w net.ipv4.udp_rmem_min=819200
   sysctl -w net.ipv4.udp_wmem_min=819200
   sysctl -w net.core.netdev_budget=600
   set +x
   #---------------------------------------------
   ```

2. Install Veritas InfoScale using the installer and select UDP as LLT protocol

   ```
   # ./installer
   ```

   The installer automatically enables the UDP Multiport feature and creates four additional sockets for each LLT link.

3. Verify that UDP multiport links are enabled

   ```
   # lltstat -nvvr configured
   ```

Enabling LLT ports in firewall

You can use any firewall tool to enable the network ports.

While enabling ports make sure that:

- No other application is using the LLT consumable network ports (50000 to 50006).
- These ports are enabled in security groups if you are installing InfoScale in cloud.

By default, LLT uses 50000 to 50001 port range for clustering and 50002 to 50006 for I/O shipping sockets.

Enabling ports using iptables

```
Ingress table:
iptables -A INPUT -p udp -m udp --dport 50000 -j ACCEPT
iptables -A INPUT -p udp -m udp --dport 50001 -j ACCEPT
iptables -A INPUT -p udp -m udp --dport 50002 -j ACCEPT
iptables -A INPUT -p udp -m udp --dport 50003 -j ACCEPT
iptables -A INPUT -p udp -m udp --dport 50004 -j ACCEPT
iptables -A INPUT -p udp -m udp --dport 50005 -j ACCEPT
iptables -A INPUT -p udp -m udp --dport 50006 -j ACCEPT


Egress table:
iptables -A OUTPUT -p udp -m udp --sport 50000 -j ACCEPT
iptables -A OUTPUT -p udp -m udp --sport 50001 -j ACCEPT
iptables -A OUTPUT -p udp -m udp --sport 50002 -j ACCEPT
iptables -A OUTPUT -p udp -m udp --sport 50003 -j ACCEPT
iptables -A OUTPUT -p udp -m udp --sport 50004 -j ACCEPT
iptables -A OUTPUT -p udp -m udp --sport 50005 -j ACCEPT
iptables -A OUTPUT -p udp -m udp --dport 50006 -j ACCEPT
```

Enable ports in the etc/llttab file

```
link eth1 udp - udp 50000 - 192.168.10.1 -
link eth2 udp - udp 50001 - 192.168.11.1 -
```

You can also use the following tunables while enabling ports.

Tunable	Description
set-udpports	Changes the port range to be used for I/O shipping if you do not want to use port range 50002 and onwards. Usage: set-udpports <initial_port_number> Example: set-udpports 60000 In this case, LLT uses the port 50000 and 50001 for clustering and 60000 and the subsequent port numbers for I/O shipping.
set-udpthreads	Specifies how many threads per socket needs to be created. Usage: set-udpthreads <number of threads per socket> Example: set-udpthreads 2
set-udpsockets	Specifies how many sockets per link needs to be created. Usage: set-udpsockets <number of sockets per link> Example: set-udpsockets 4

Disabling the UDP multiport feature

The UDP multiport feature is enabled by default. You can manually disable it by setting the value of the **LLT_UDP_MULTIPORT** variable to **zero**. This variable is defined in the `/etc/sysconfig/llt` file.

That is, in the `/etc/sysconfig/llt` file, set **LLT_UDP_MULTIPORT= 0**

Configuring LLT over TCP

This appendix includes the following topics:

Using the TCP layer for LLT

VCS provides the option of using LLT over the Transmission Control Protocol (TCP) layer for clusters using wide-area networks and routers. TCP makes LLT packets routable and thus able to span longer distances more economically.

Manually configuring LLT over TCP using IPv4

The following checklist is to configure LLT over TCP:

- Set the broadcast address in /etc/llttab explicitly depending on the subnet for the link.
  See "Broadcast address in the /etc/llttab file" on page 308.
- Make sure that the NIC has an IP address that is configured before configuring LLT.
- Make sure the IP address in the /etc/llttab files is consistent with the IP address of the network interfaces.
- Make sure that the link has a unique not well-known TCP port.
  See "Selecting TCP ports" on page 310.
- Set the broadcast address correctly for direct-attached (non-routed) link.

See “Sample configuration: direct-attached link” on page 311.

- For the link that cross an IP router, disable broadcast features and specify the IP address of the link manually in the /etc/llttab file.
  See “Sample configuration: link crossing IP routers” on page 312.

Broadcast address in the /etc/llttab file

The broadcast address is set explicitly for the link in the following example.

- Display the content of the /etc/llttab file on the first node sys1:

```
sys1 # cat /etc/llttab

set-node sys1
set-cluster 1
link link1 udp - tcp  50000  -  192.168.9.1 192.168.9.255
```

- Display the content of the /etc/llttab file on the second node sys2:

```
sys2 # cat /etc/llttab

set-node sys2
set-cluster 1
link link1 udp - tcp  50000  -  192.168.9.2 192.168.9.255
```

The link command in the /etc/llttab file

Review the link command information in this section for the /etc/llttab file. See the following information for sample configurations:

- See “Sample configuration: direct-attached link” on page 311.
- See “Sample configuration: link crossing IP routers” on page 312.

Table D-1 describes the fields of the link command that are shown in the /etc/llttab file examples. Note that some of the fields differ from the command for standard LLT links.

Table D-1 Field description for link command in /etc/llttab

Field	Description
tag-name	A unique string that is used as a tag by LLT; for example link1, link2,....

Table D-1 Field description for link command in /etc/llttab *(continued)*

Field	Description
device	The device path of the TCP protocol; for example tcp. A place holder string. Linux does not have devices for protocols. So this field is ignored.
node-range	Nodes using the link. "-" indicates all cluster nodes are to be configured for this link.
link-type	Type of link; must be "tcp" for LLT over TCP.
tcp-port	Unique TCP port in the range of 49152-65535 for the link. See "Selecting TCP ports" on page 310.
MTU	"-" is the default, which has a value of 8192. The value may be increased or decreased depending on the configuration. Use the `lltstat -l` command to display the current value.
IP address	IP address of the link on the local node.
bcast-address	■ For clusters with enabled broadcasts, specify the value of the subnet broadcast address. ■ "-" is the default for clusters spanning routers.

The set-addr command in the /etc/llttab file

The `set-addr` command in the /etc/llttab file is required when the broadcast feature of LLT is disabled, such as when LLT must cross IP routers.

See "Sample configuration: link crossing IP routers" on page 312.

Table D-2 describes the fields of the `set-addr` command.

Table D-2 Field description for set-addr command in /etc/llttab

Field	Description
node-id	The node ID of the peer node; for example, 0.
link tag-name	The string that LLT uses to identify the link; for example link1, link2,....
address	IP address assigned to the link for the peer node.

Selecting TCP ports

When you select a TCP port, select an available 16-bit integer from the range that follows:

- Use available ports in the private range 49152 to 65535
- Do not use the following ports:
  - Ports from the range of well-known ports, 0 to 1023
  - Ports from the range of registered ports, 1024 to 49151
- Ports already used by some other application

To check which ports are defined as defaults for a node, examine the file /etc/services. You should also use the `netstat` command to list the TCP ports currently in use. For example:

```
# netstat -au | more
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address Foreign Address         State
tcp        0      0 *:32768                 *:*
udp        0      0 *:956                   *:*
tcp        0      0 *:tftp                  *:*
tcp        0      0 *:sunrpc                *:*
udp        0      0 *:ipp                   *:*
```

Look in the TCP section of the output; the TCP or UDP ports that are listed under Local Address are already in use. If a port is listed in the /etc/services file, its associated name is displayed rather than the port number in the output.

Configuring the netmask for LLT

Run the following command and answer the prompt to set the netmask:

```
# ifconfig interface_name netmask netmask
```

For example:

- For the first network interface on the node sys1:

  ```
  IP address=192.168.9.1, Broadcast address=192.168.9.255,
  Netmask=255.255.255.0
  ```

  For the first network interface on the node sys2:

  ```
  IP address=192.168.9.2, Broadcast address=192.168.9.255,
  Netmask=255.255.255.0
  ```

- For the second network interface on the node sys1:

  ```
  IP address=192.168.10.1, Broadcast address=192.168.10.255,
  Netmask=255.255.255.0
  ```

  For the second network interface on the node sys2:

  ```
  IP address=192.168.10.2, Broadcast address=192.168.10.255,
  Netmask=255.255.255.0
  ```

Configuring the broadcast address for LLT

An example of a typical /etc/llttab file:

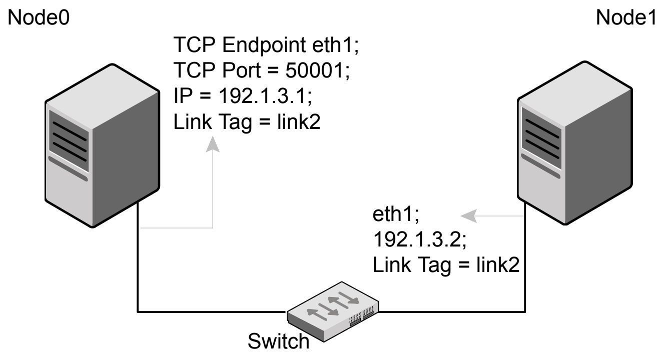
```
# cat /etc/llttab
set-node nodexyz
set-cluster 100

link link1 udp - tcp 50000 - 192.168.30.1 192.168.30.255
```

Sample configuration: direct-attached link

Figure D-1 depicts a typical configuration of direct-attached link employing LLT over TCP.

Figure D-1 A typical configuration of direct-attached link that use LLT over TCP



The configuration that the /etc/llttab file for Node 0 represents has directly attached crossover link. It might also have the link that is connected through a hub or switch. These link does not cross routers.

LLT sends broadcast requests to peer nodes to discover their addresses. So the addresses of peer nodes do not need to be specified in the /etc/llttab file using the `set-addr` command. For direct attached link, you do need to set the broadcast address of the link in the /etc/llttab file. Verify that the IP address and broadcast address are set correctly by using the `ifconfig -a` command.

```
set-node Node0
set-cluster 1
#configure Links
#link tag-name
    device
    node-range
    link-type
    tcp port
    MTU \
IP-address
    bcast-address
link link1 udp - tcp 50000 - 192.1.2.1 192.1.2.255
```

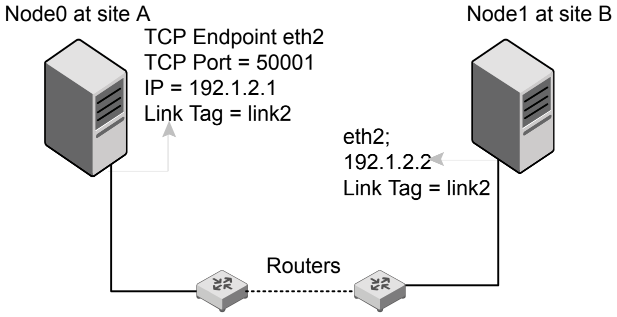
The file for Node 1 resembles:

```
set-node Node1
set-cluster 1
#configure Links
#link tag-name
    device
    node-range
    link-type
    tcp port
    MTU \
IP-address
    bcast-address
link link1 udp - tcp 50000 - 192.1.2.2 192.1.2.255
```

Sample configuration: link crossing IP routers

Figure D-2 depicts a typical configuration of link crossing an IP router employing LLT over TCP. The illustration shows two nodes of a four-node cluster.

Figure D-2 A typical configuration of link crossing an IP router



The configuration that the following `/etc/llttab` file represents for Node 1 has a link crossing IP routers. Notice that IP address is shown for the link on each peer node. In this configuration broadcast is disabled. Hence, the broadcast address does not need to be set in the `link` command of the `/etc/llttab` file.

```
set-node Node1
set-cluster 1
link link1 udp - tcp 50000 - 192.1.3.1 -


#set address of each link for all peer nodes in the cluster
#format: set-addr node-id
    link tag-name
    address
set-addr        0 link1 192.1.1.1
set-addr        2 link1 192.1.5.2
set-addr        3 link1 192.1.7.3

#disable LLT broadcasts
set-bcasthb     0
set-arp         0
```

The `/etc/llttab` file on Node 0 resembles:

```
set-node Node0
set-cluster 1
```

```
link link1 udp - tcp 50000 - 192.1.1.1 -

#set address of each link for all peer nodes in the cluster
#format: set-addr node-id
    link tag-name
    address
set-addr        1 link1 192.1.3.1
set-addr        2 link1 192.1.5.2
set-addr        3 link1 192.1.7.3


#disable LLT broadcasts
set-bcasthb     0
set-arp         0
```

Manually configuring LLT over TCP using IPv6

Use the following checklist when configuring LLT over TCP with IPv6:

- Make sure that each NIC has an IPv6 address which is configured before configuring LLT.
- Make sure the IPv6 addresses in the /etc/llttab files are consistent with the IPv6 addresses of the network interfaces.
- Make sure that each link has a unique not well-known TCP port.
  See "Selecting TCP ports" on page 316.
- For the links that cross an IP router, disable multicast features and specify the IPv6 address of each link manually in the /etc/llttab file.
  See "Sample configuration: links crossing IP routers" on page 318.
- For the local TCP6 link configured in /etc/llttab, add "set-addr" for all peer nodes with same link tag containing the peer nodes link specific IPv6 address.
  See "The set-addr command in the /etc/llttab file" on page 315.
- When LLT is configured over TCP, in any case, only one TCP link, either IPV4 or IPv6, should present in the configuration.

The link command in the /etc/llttab file

Review the link command information in this section for the /etc/llttab file. See the following information for sample configurations:

- See "Sample configuration: direct-attached links" on page 316.
- See "Sample configuration: links crossing IP routers" on page 318.

Note that some of the fields in Table D-3 differ from the command for standard LLT links.

Table D-3 describes the fields of the link command that are shown in the /etc/llttab file examples.

Table D-3 Field description for link command in /etc/llttab

Field	Description
tag-name	A unique string that is used as a tag by LLT; for example link1, link2,....
device	The device name of the TCP protocol; for example tcp6.
node-range	Nodes using the link. "-" indicates all cluster nodes are to be configured for this link.
link-type	Type of link; must be "tcp6" for LLT over TCP.
tcp-port	Unique TCP port in the range of 49152-65535 for the link. See "Selecting TCP ports" on page 316.
MTU	"-" is the default, which has a value of 8192. The value may be increased or decreased depending on the configuration. Use the `lltstat -l` command to display the current value.
IPv6 address	IPv6 address of the link on the local node.
mcast-address	"-" is the default for clusters spanning routers.
set-addr	IPv6 address of the peer node that has the same link tag as local link.

The set-addr command in the /etc/llttab file

The `set-addr` command in the `/etc/llttab` file is required when the multicast feature of LLT is disabled, such as when LLT must cross IP routers.

See "Sample configuration: links crossing IP routers" on page 318.

Table D-4 describes the fields of the set-addr command.

Table D-4 Field description for set-addr command in /etc/llttab

Field	Description
node-id	The ID of the peer node; for example, 0.

Table D-4 Field description for set-addr command in /etc/llttab *(continued)*

Field	Description
link tag-name	The string that LLT uses to identify the link; for example link1, link2,....
address	IPv6 address assigned to the link for the peer node.

Selecting TCP ports

When you select a TCP port, select an available 16-bit integer from the range that follows:

- Use available ports in the private range 49152 to 65535
- Do not use the following ports:
  - Ports from the range of well-known ports, 0 to 1023
  - Ports from the range of registered ports, 1024 to 49151
- Ports already used by some other application

To check which ports are defined as defaults for a node, examine the file `/etc/services`. You should also use the `netstat` command to list the TCP ports currently in use. For example:

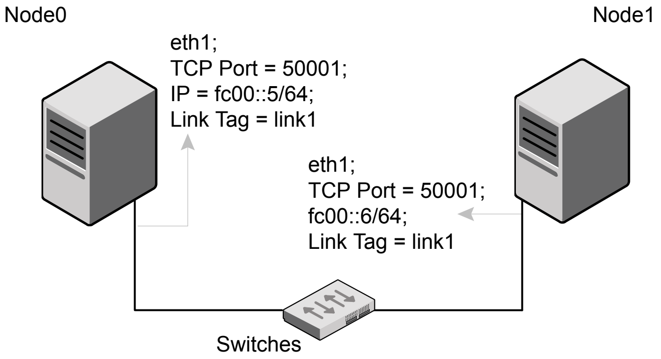
```
# netstat -au | more
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address Foreign Address         State
tcp        0      0 *:32768                *:*
udp        0      0 *:956                  *:*
tcp        0      0 *:tftp                 *:*
tcp        0      0 *:sunrpc               *:*
udp        0      0 *:ipp                  *:*
```

Look in the TCP section of the output; the TCP ports that are listed under Local Address are already in use. If a port is listed in the `/etc/services` file, its associated name is displayed rather than the port number in the output.

Sample configuration: direct-attached links

Figure D-3 depicts a typical configuration of direct-attached links employing LLT over TCP.

Figure D-3 A typical configuration of direct-attached links that use LLT over TCP



The configuration that the `/etc/llttab` file for Node 0 represents has directly attached crossover links. It might also have the links that are connected through a hub or switch. These links do not cross routers.

```
set-node Node0
set-cluster 1234
#configure Links
#link tag-name
    device
    node-range
    link-type
    tcp port
    MTU \
IP-address
    mcast-address
link link1 udp6 - tcp6 50000 - fc00::5 -
set-addr 1 link1 fc00::6
```

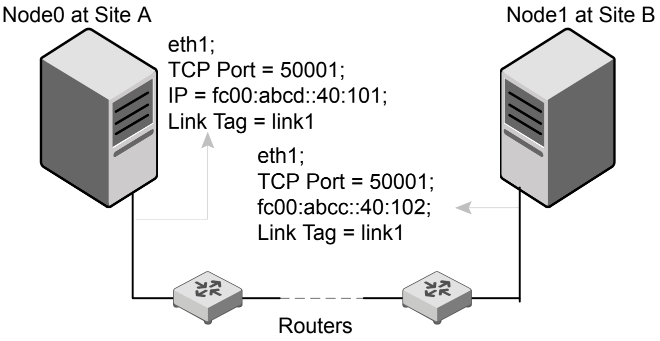
The file for Node 1 resembles:

```
set-node Node1
set-cluster 1234
#configure Links
#link tag-name
    device
    node-range
    link-type
    tcp port
    MTU \
IP-address
    mcast-address
link link1 udp6 - tcp6 50000 - fc00::6 -
set-addr 0 link1 fc00::5
```

Sample configuration: links crossing IP routers

Figure D-4 depicts a typical configuration of links crossing an IP router employing LLT over TCP. The illustration shows two nodes of a four-node cluster.

Figure D-4 A typical configuration of links crossing an IP router



The configuration that the following `/etc/llttab` file represents for Node 1 has links crossing IP routers. Notice that IPv6 addresses are shown for each link on each peer node. In this configuration multicasts are disabled.

```
set-node Node1
set-cluster 1234

link link1 udp6 - tcp6 50000 - fc00::6

#set address of each link for all peer nodes in the cluster
#format: set-addr node-id
    link tag-name
    address
set-addr 0 link1 fc00::5
```

```
set-addr 2 link1 fc00::7
set-addr 3 link1 fc00::8

#disable LLT multicasts
set-bcasthb     0
set-arp         0
```

The /etc/llttab file on Node 0 resembles:

```
set-node Node0
set-cluster 1

link link1 udp6 - tcp6 50000 - fc00::5

#set address of each link for all peer nodes in the cluster
#format: set-addr node-id
    link tag-name
    address
set-addr 1 link1 fc00::6
set-addr 2 link1 fc00::7
set-addr 3 link1 fc00::8


#disable LLT multicasts
set-bcasthb     0
set-arp         0
```

LLT over TCP sample /etc/llttab

The following is a sample of LLT over TCP in the etc/llttab file.

```
set-node sys1
set-cluster clus1
link eth1 udp - tcp 50000 - 192.168.10.1 -
set-addr 1 eth1 192.168.10.2
set-bcasthb 0
set-arp 0
```

Migrating LLT links from IPv4 to IPv6 or dual-stack

This appendix includes the following topics:

About migrating the LLT links from IPv4 to IPv6 or to a dual-stack network

InfoScale supports pure IPv4, pure IPv6, as well as dual-stack deployments.

You can consider migrating a network from a IPv4 to IPv6 in the following scenarios:

- When LLT is configured over UDP using IPv4
- When LLT is configured over TCP using IPv4
- When adding a new node to an existing cluster

Perform the migration one node at a time so that you can easily rollback in case you face issues with the migration.

To migrate to IPv6, do the following:

1. Review the current configuration
2. Meet the prerequisites for migration
3. Perform the actual migration
4. Retain or remove the IPv4 link

When LLT is configured over TCP, only one TCP link, either IPV4 or IPv6, should be present in the configuration. Therefore, you should add a non-TCP IPv4 link for the migration. This ensures that the additional non-TCP link continues to be operational using IPv4, while the other link is being migrated to IPv6.

Review the current configuration

Before you migrate your existing LLT configuration from IPv4 to IPv6, review your current configuration using either the `cat /etc/llttab` or the `lltstat -l` command.

These commands provide the following details about your current configuration:

- Number of LLT links configured
- IP addresses and ports configured for the LLT links
- Link-tag names for the configured links

Meet the prerequisites for migration

Before you start the migration, ensure that the following prerequisites are met:

- The cluster is running with LLT configured over UDP or TCP using IPv4.
- The broadcast address is set explicitly for each link.
- For nodes in different subnets, the netmask is set so that the nodes can access the subnets in use.
- Each NIC has an IPv6 address that is configured before configuring LLT.
- Each link has a unique not well-known UDP or TCP port.
- IPv6 is configured on all the nodes before you perform the migration. That is, the nodes must have both IPv4 and IPv6 addresses configured.

Adding a new node to an existing cluster

If you want to add a node to an existing cluster that is configured using IPv4, and want to configure the new LLT links using IPv6, perform the following steps:

Task	Reference
Upgrade the existing cluster to IPv6 configuration	See "Migrating LLT links to IPv6 or dual-stack when LLT is configured over UDP using IPv4" on page 322. See "Migrating LLT links to IPv6 or dual-stack when LLT is configured over TCP using IPv4" on page 323.
Add the new node to an existing cluster	See "Adding a node to a single-node cluster" on page 231. See "Adding nodes using the VCS installer" on page 239. See "Manually adding a node to a cluster" on page 242.

Migrating LLT links to IPv6 or dual-stack when LLT is configured over UDP using IPv4

To dynamically migrate from IPv4 to IPv6 configuration:

1. Make sure that the LLT links are operational.

   ```
   # lltstat -nvv configured
   ```

2. Make sure that the IPv6 address is plumbed on all the nodes and is pingable from IPv6 interface addresses of the peer nodes.

3. On all the nodes, add the LLT TCP link with the newly plumbed IPv6 address dynamically.

   ```
   # lltconfig -t devtag -d device -b tcp6 -s port [-m mtu] -I IPaddr
   [-B mcast] [-l] [-Q]
   ```

   where,

   devtag is different than the existing device tag used in `llttab`

   device is the interface name used for the IPv6 link

4. Verify that the TCP6 link is added properly and the links are operational.

   ```
   # lltstat -l
   ```

   ```
   # lltstat -nvv configured
   ```

5 Optionally, disable and unlink the existing IPv4 link in llttab.

```
# lltconfig -L disable -t linkTagName
```

```
# lltconfig -u linkagName
```

6 Verify whether all the high priority LLT links are now configured with IPv6 address.

```
lltstat -nvv configured
```

7 Update llttab with the latest link configurations.

If the link configurations are not updated, the migration changes are lost after reboot or stack restart.

Migrating LLT links to IPv6 or dual-stack when LLT is configured over TCP using IPv4

Before you migrate from IPv4 to IPv6, ensure that the cluster is running with LLT configured over TCP using IPv4.

To dynamically migrate from IPv4 to IPv6 configuration:

1 Make sure that the LLT link is operational.

```
# lltstat -nvv configured
```

2 Make sure that the IPv6 address is plumbed on all the nodes and is pingable from the IPv6 interface addresses of the peer nodes.

3 On one of the interfaces, add one non-TCP (UDP) IPv4 link.

LLT over TCP configuration allows only one TCP link. Therefore, you should add a non-TCP IPv4 link for the migration. Doing so ensures that the additional non-TCP link continues to be operational, while the other link is being migrated to IPv6.

Use the following command to add the LLT UDP link with the newly plumbed IPv4 address dynamically:

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

where,

devtag is different than existing devtag used in llttab

device is the interface name used for the IPv4 link

4 Verify that the non-TCP IPv4 link is added properly and is operational.

```
# lltstat -l
```

```
# lltstat -nvv configured
```

5 Disable the existing TCP IPv4 link.

```
# lltconfig -L disable -t linkTagName
```

6 Unlink the disabled TCP IPv4 LLT link.

```
# lltconfig -u linkTagName
```

7 Verify whether one of the high priority LLT links is now configured with non-TCP IPv4 address.

```
# lltstat -l
```

```
# lltstat -nvv configured
```

8 Update `/etc/llttab` with `set-addr` for all peer nodes with the IPv6 addresses that to be used for the TCPv6 links.

9 Add the TCP LLT link with the newly plumbed IPv6 address dynamically.

```
# lltconfig -t devtag -d device -b tcp6 -s port [-m mtu] -I IPaddr
[-B mcast] [-l] [-Q]
```

where,

devtag is same as the existing devtag of the TCP link used in llttab

device is the interface name used for the IPv6 link

10 Update `/etc/llttab` with the latest link configurations.

If the link configurations are not updated, the migration changes are lost after reboot.

Using LLT over RDMA

This appendix includes the following topics:

Using LLT over RDMA

This section describes how LLT works with RDMA, lists the hardware requirements for RDMA, and the procedure to configure LLT over RDMA.

About RDMA over RoCE or InfiniBand networks in a clustering environment

Remote direct memory access (RDMA) is a direct memory access capability that allows server to server data movement directly between application memories with minimal CPU involvement. Data transfer using RDMA needs RDMA-enabled network cards and switches. Networks designed with RDMA over Converged Ethernet (RoCE) and InfiniBand architecture support RDMA capability. RDMA provides fast interconnect between user-space applications or file systems between nodes over these networks. In a clustering environment, RDMA capability allows applications on separate nodes to transfer data at a faster rate with low latency and less CPU usage.

See “How LLT supports RDMA capability for faster interconnects between applications” on page 326.

How LLT supports RDMA capability for faster interconnects between applications

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

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

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

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

See “Using LLT over RDMA: supported use cases ” on page 327.

Using LLT over RDMA: supported use cases

You can configure the LLT over RDMA capability for the following use cases:

- Storage Foundation Smart IO feature on flash storage devices: The Smart IO feature provides file system caching on flash devices for increased application performance by reducing IO bottlenecks. It also reduces IO loads on storage controllers as the Smart IO feature meets most of the application IO needs. As the IO requirements from the storage array are much lesser, you require lesser number of servers to maintain the same IO throughput.
- Storage Foundation IO shipping feature: The IO shipping feature in Storage Foundation Cluster File System HA (SFCFSHA) provides the ability to ship IO data between applications on peer nodes without service interruption even if the IO path on one of the nodes in the cluster goes down.
- Storage Foundation Flexible Storage Sharing feature : The Flexible Storage Sharing feature in cluster volume manager allows network shared storage to co-exist with physically shared storage. It provides server administrators the ability to provision clusters for Storage Foundation Cluster File System HA (SFCFSHA) and Storage Foundation for Oracle RAC (SFRAC) or SFCFSHA applications without requiring physical shared storage.

Both Cluster File System (CFS) and Cluster Volume Manager (CVM) are clients of LLT and GAB. These clients use LLT as the transport protocol for data transfer between applications on nodes. Using LLT data transfer over an RDMA network boosts performance of file system data transfer and IO transfer between nodes.

To enable RDMA capability for faster application data transfer between nodes, you must install RDMA-capable network interface cards, RDMA-supported network switches, configure the operating system for RDMA, and configure LLT.

Ensure that you select RDMA-supported hardware and configure LLT to use RDMA functionality.

See “Choosing supported hardware for LLT over RDMA” on page 328.

See “Configuring LLT over RDMA” on page 327.

Configuring LLT over RDMA

This section describes the required hardware and configuration needed for LLT to support RDMA capability. The high-level steps to configure LLT over RDMA are as follows:

Table F-1 lists the high-level steps to configure LLT over RDMA.

Step	Action	Description
Choose supported hardware	Choose RDMA capable network interface cards (NICs), network switches, and cables.	See "Choosing supported hardware for LLT over RDMA" on page 328.
Check the supported operating system	Verify that it is a supported Linux operating system.	All supported Linux flavors
Install RDMA, InfiniBand or Ethernet drivers and utilities	Install the packages to access the RDMA, InfiniBand or Ethernet drivers and utilities.	See "Installing RDMA, InfiniBand or Ethernet drivers and utilities" on page 329.
Configure RDMA over an Ethernet network	Load RDMA and Ethernet drivers.	See "Configuring RDMA over an Ethernet network" on page 330.
Configuring RDMA over an InfiniBand network	Load RDMA and InfiniBand drivers.	See "Configuring RDMA over an InfiniBand network" on page 332.
Tune system performance	Tune CPU frequency and boot parameters for systems.	See "Tuning system performance" on page 336.
Configure LLT manually	Configure LLT to use RDMA capability. Alternatively, you can use the installer to automatically configure LLT to use RDMA.	See "Manually configuring LLT over RDMA" on page 338.
Verify LLT configuration	Run LLT commands to test the LLT over RDMA configuration.	See "Verifying LLT configuration" on page 342.

Choosing supported hardware for LLT over RDMA

To configure LLT over RDMA you need to use the hardware that is RDMA enabled.

Table F-2

Hardware	Supported types	Reference
Network card	Mellanox-based Host Channel Adapters (HCAs) (VPI, ConnectX, ConnectX-2 and 3)	For detailed installation information, refer to the hardware vendor documentation.

Table F-2 *(continued)*

Hardware	Supported types	Reference
Network switch	Mellanox, InfiniBand switches Ethernet switches must be Data Center Bridging (DCB) capable	For detailed installation information, refer to the hardware vendor documentation.
Cables	Copper and Optical Cables, InfiniBand cables	For detailed installation information, refer to the hardware vendor documentation.

Warning: When you install the Mellanox NIC for using RDMA capability, do not install Mellanox drivers that come with the hardware. LLT uses the Mellanox drivers that are installed by default with the Linux operating system. LLT might not be configurable if you install Mellanox drivers provided with the hardware.

Installing RDMA, InfiniBand or Ethernet drivers and utilities

Install the following RPMs to get access to the required RDMA, InfiniBand or Ethernet drivers and utilities. Note that the rpm version of the RPMs may differ for each of the supported Linux flavors.

Veritas does not support any external Mellanox OFED packages. The supported packages are listed in this section.

Veritas recommends that you use the Yellowdog Updater Modified (yum) package management utility to install RPMs on RHEL systems and use Zypper, a command line package manager, on SUSE systems.

Note: Install the OpenSM package only if you configure an InfiniBand network. All other packages are required with both InfiniBand and Ethernet networks.

Table F-3 lists the drivers and utilities required for RDMA, InfiniBand or Ethernet network.

Packages	RHEL	SUSE
Userland device drivers for RDMA operations	■ `libmthca` ■ `libmlx4` ■ `rdma` ■ `librdmacm-utils`	■ `libmthca-rdmav2` ■ `libmlx4-rdmav2` ■ `ofed` ■ `librdmacm`

Table F-3 lists the drivers and utilities required for RDMA, InfiniBand or Ethernet network. *(continued)*

Packages	RHEL	SUSE
OpenSM related package (InfiniBand only)	■ `opensm` ■ `opensm-libs` ■ `libibumad`	■ `opensm` ■ `libibumad3`
InfiniBand troubleshooting and performance tests	■ `Ibutils` ■ `infiniband-diags` ■ `Perftest`	■ `Ibutils` ■ `infiniband-diags`
libibverbs packages for userland InfiniBand operations	■ `libibverbs-devel` ■ `libibverbs-utils`	■ `libibverbs`

Configuring RDMA over an Ethernet network

Configure the RDMA and Ethernet drivers so that LLT can use the RDMA capable hardware.

See "Enable RDMA over Converged Ethernet (RoCE)" on page 330.

See "Configuring RDMA and Ethernet drivers" on page 331.

See "Configuring IP addresses over Ethernet Interfaces" on page 331.

Enable RDMA over Converged Ethernet (RoCE)

The following steps are applicable only on a system installed with RHEL Linux or supported RHEL-compatible distributions. On SUSE Linux, the RDMA is enabled by default.

1. Make sure that the SFHA stack is stopped and the LLT and GAB modules are not loaded.

   See "Starting and stopping processes for the Veritas InfoScale products " on page 363.

2. Skip this step if you are on a RHEL 7 or supported RHEL-compatible distributions. Alternatively, create or modify the `/etc/modprobe.d/mlx4.conf` configuration file and add the value `options mlx4_core hpn=1` to the file. This enables RDMA over Converged Ethernet (RoCE) in Mellanox drivers (installed by default with the operating system).

3. Verify whether the Mellanox drivers are loaded.

   ```
   # lsmod | grep mlx4_en
   ```

   ```
   # lsmod | grep mlx4_core
   ```

4. Unload the Mellanox drivers if the drivers are loaded.

   ```
   # rmmod mlx4_ib
   ```

   ```
   # rmmod mlx4_en
   ```

   ```
   # rmmod mlx4_core
   ```

Configuring RDMA and Ethernet drivers

Load the Mellanox drivers that are installed by default with the operating system and enable the RDMA service.

1. (RHEL and supported RHEL-compatible distributions only) Load the Mellanox drivers.

   ```
   # modprobe mlx4_core
   ```

   ```
   # modprobe mlx4_ib
   ```

   ```
   # modprobe mlx4_en
   ```

2. Enable RDMA service on the Linux operating system.

   On RHEL Linux: `# chkconfig --level 235 rdma on`

   On SUSE Linux: `# chkconfig --level 235 openibd on`

Configuring IP addresses over Ethernet Interfaces

Perform the following steps to configure IP addresses over the network interfaces which you plan to configure under LLT. These interfaces must not be aggregated interfaces.

1. Configure IP addresses using Linux `ifconfig` command. Make sure that the IP address for each link must be from a different subnet.

   Typical private IP addresses that you can use are:

   Node0:

   link0: 192.168.1.1

   link1: 192.168.2.1

   Node1:

   link0: 192.168.1.2

   link1: 192.168.2.2

2. Run IP ping test between nodes to ensure that there is network level connectivity between nodes.

3. Configure IP addresses to start automatically after the system restarts or reboots by creating a new configuration file or by modifying the existing file.

   - On RHEL or supported RHEL-compatible distributions, modify the `/etc/sysconfig/network-scripts/` directory by modifying the `ifcfg-eth` (Ethernet) configuration file.
   - On SUSE, modify the `/etc/sysconfig/network/` by modifying the `ifcfg-eth` (Ethernet) configuration file.
     For example, for an Ethernet interface eth0, create the `ifcfg-eth0` file with values for the following parameters.

```
DEVICE=eth0
BOOTPROTO=static
IPADDR=192.168.27.1
NETMASK=255.255.255.0
NETWORK=192.168.27.0
BROADCAST=192.168.27.255
NM_CONTROLLED=no # This line ensures IPs are plumbed correctly after bootup
ONBOOT=yes
STARTMODE='auto' # This line is only for SUSE
```

Configuring RDMA over an InfiniBand network

While configuring RDMA over an InfiniBand network, you need to configure the InfiniBand drivers, configure the OpenSM service, and configure IP addresses for the InfiniBand interfaces.

See “Configuring RDMA and InfiniBand drivers” on page 333.

See “ Configuring the OpenSM service” on page 334.

See “Configuring IP addresses over InfiniBand Interfaces” on page 335.

Configuring RDMA and InfiniBand drivers

Configure the RDMA and InfiniBand drivers so that LLT can use the RDMA capable hardware.

1. Ensure that the following RDMA and InfiniBand drivers are loaded. Use the `lsmod` command to verify whether a driver is loaded.

   The InfiniBand interfaces are not visible by default until you load the InfiniBand drivers. This procedure is only required for initial configuration.

   ```
   # modprobe rdma_cm

   # modprobe rdma_ucm

   # modprobe mlx4_en

   # modprobe mlx4_ib

   # modprobe ib_mthca

   # modprobe ib_ipoib

   # modprobe ib_umad
   ```

2. Load the drivers at boot time by appending the configuration file on the operating system.

   On RHEL and SUSE Linux, append the `/etc/rdma/rdma.conf` and `/etc/infiniband/openib.conf` files respectively with the following values:

   ```
   ONBOOT=yes

   RDMA_UCM_LOAD=yes

   MTHCA_LOAD=yes

   IPOIB_LOAD=yes

   SDP_LOAD=yes

   MLX4_LOAD=yes

   MLX4_EN_LOAD=yes
   ```

3. Enable RDMA service on the Linux operating system.

   On RHEL Linux:

   ```
   # chkconfig --level 235 rdma on
   ```

   On SUSE Linux:

   ```
   # chkconfig --level 235 openibd on
   ```

Configuring the OpenSM service

OpenSM is an InfiniBand compliant Subnet Manager and Subnet Administrator, which is required to initialize the InfiniBand hardware. In the default mode, OpenSM scans the IB fabric, initializes the hardware, and checks the fabric occasionally for changes.

For InfiniBand network, make sure to configure subnet manager if you have not already configured the service.

1 Modify the OpenSM configuration file if you plan to configure multiple links under LLT.

  On RHEL, update the `/etc/sysconfig/opensm` file.

2 Start OpenSM.

  For RHEL 7, SLES 12, and supported RHEL distributions:, run **`# systemctl start opensm`**

  For earlier versions of RHEL, SLES and supported RHEL distributions, run **`# /etc/init.d/opensm start`**

3 Enable Linux service to start OpenSM automatically after restart.

  For RHEL 7 and supported RHEL distributions, **`# systemctl enable opensm`**

  For SLES 12, **`# systemctl enable opensmd`**

Configuring IP addresses over InfiniBand Interfaces

Perform the following steps to configure IP addresses over the network interfaces which you plan to configure under LLT. These interfaces must not be aggregated interfaces.

1 Configure IP addresses using Linux `ifconfig` command. Make sure that the IP address for each link must be from a different subnet.

  Typical private IP addresses that you can use are: **192.168.12.1, 192.168.12.2, 192.168.12.3** and so on.

2 Run the InfiniBand ping test between nodes to ensure that there is InfiniBand level connectivity between nodes.

  - On one node, start the ibping server.

    ```
    # ibping -S
    ```

  - On the node, get the GUID of an InfiniBand interface that you need to ping from another node.

    ```
    # ibstat


    CA 'mlx4_0'
    Number of ports: 2
     --
    ```

```
Port 1:
State: Active
---
Port GUID: 0x0002c90300a02af1
Link layer: InfiniBand
```

- Ping the peer node by using its GUID.

  ```
  # ibping -G 0x0002c90300a02af1
  ```

  Where, *0x0002c90300a02af1* is the GUID of the server.

3 Configure IP addresses automatically after restart by creating a new configuration file or by modifying the existing file.

- On RHEL, modify the `/etc/sysconfig/network-scripts/` directory by modifying the `ifcfg-ibX` (InfiniBand) configuration file.
- On SUSE, modify the `/etc/sysconfig/network/` by modifying the `ifcfg-ibX` (InfiniBand) configuration file.

```
For example, for an Infiniband interface ib0,
create ifcfg-ib0 file with values for the following parameters.

DEVICE=ib0
BOOTPROTO=static
IPADDR=192.168.27.1
NETMASK=255.255.255.0
NETWORK=192.168.27.0
BROADCAST=192.168.27.255
NM_CONTROLLED=no # This line ensures IPs are plumbed correctly
after bootup and the Network manager does not interfere
with the interfaces
ONBOOT=yes
STARTMODE='auto' # This line is only for SUSE
```

Tuning system performance

Run IP ping test to ensure that systems are tuned for the best performance. The latency should be less than 30us, if not then your system may need tuning. However, the latency may vary based on your system configuration.

To tune your system, perform the following steps. For additional tuning, follow the performance tuning guide from Mellanox.

Performance Tuning Guidelines for Mellanox Network Adapters

Tuning the CPU frequency

To tune the CPU frequency of a system, perform the following steps:

1. Verify whether the CPU frequency is already tuned.

   ```
   # cat /proc/cpuinfo | grep Hz


   model name      : Intel(R) Xeon(R) CPU E5-2643 0 @ 3.30GHz
   cpu MHz         : 3300.179
   ```

2. If the CPU frquency displayed by the `cpu MHz` and `model name` attribute is the same, then the CPU frequency is already tuned. You can skip the next steps.

   If the CPU frequency displayed by the `cpu Mhz` and `model name` attribute is not the same, then follow the next steps to tune the frequency.

3. Go to system console and restart the system.
4. Press F11 to enter into BIOS settings.
5. Go to BIOS menu > Launch System setup > BIOS settings > System Profile Settings > System Profile > Max performance.

   The menu options might vary with system type.

Tuning the boot parameter settings

To tune the boot parameter settings, perform the following steps.

1. In the `/boot/grub/grub.conf` file or any other boot loader configuration file, ensure that the value of the `intel_iommu` is set to **off**.
2. Append the `/boot/grub/grub.conf` file or any other boot loader configuration file with the following parameters if they are not listed in the configuration file.

   ```
   intel_idle.max_cstate=0 processor.max_cstate=1
   ```

3. Restart the system.

On RHEL 7 and supported RHEL-compatible distributions:

1. In the `/etc/default/grub` file, append the GRUB_CMDLINE_LINUX variable with intel_idle.max_cstate=0 processor.max_cstate=1
2. After `/etc/default/grub` is modified, run the following command:

   ```
   grub2-mkconfig -o /boot/grub2/grub.cfg
   ```

3. Restart the system.

Manually configuring LLT over RDMA

You can automatically configure LLT to use RDMA using the installer. To manually configure LLT over RDMA follow the steps that are given in this section.

The following checklist is to configure LLT over RDMA:

- Make sure that the LLT private links are on separate subnets. Set the broadcast address in `/etc/llttab` explicitly depending on the subnet for each link.

  See "Broadcast address in the `/etc/llttab` file" on page 338.
- Make sure that each RDMA enabled NIC (RNIC) over an InfiniBand or Ethernet network has an IP address that is configured before configuring LLT.
- Make sure that the IP addresses in the `/etc/llttab` files are consistent with the IP addresses of the network interfaces (InfiniBand or Ethernet network interfaces).
- Make sure that each link has a unique and a private IP range for the UDP port.

  See "Selecting UDP ports" on page 339.
- See the sample comfiguration for direct-attached (non-routed) links.

  See "Sample configuration: direct-attached links" on page 341.

Broadcast address in the `/etc/llttab` file

The broadcast address is set explicitly for each link in the following example.

- Display the content of the `/etc/llttab` file on the first node sys1:

  ```
  sys1 # cat /etc/llttab

  set-node sys1
  set-cluster 1
  link link1 udp - rdma  50000  -  192.168.9.1 192.168.9.255
  link link2 udp - rdma  50001  -  192.168.10.1 192.168.10.255
  ```

  Verify the subnet mask using the ifconfig command to ensure that the two links are on separate subnets.
- Display the content of the `/etc/llttab` file on the second node sys2:

  ```
  sys2 # cat /etc/llttab

  set-node sys2
  set-cluster 1
  link link1 udp - rdma  50000  -  192.168.9.2 192.168.9.255
  link link2 udp - rdma  50001  -  192.168.10.2 192.168.10.255
  ```

Verify the subnet mask using the ifconfig command to ensure that the two links are on separate subnets.

The link command in the `/etc/llttab` file

Review the link command information in this section for the `/etc/llttab` file. See the following information for sample configurations:

-

Table F-4 describes the fields of the link command that are shown in the `/etc/llttab` file examples. Note that some of the fields differ from the command for standard LLT links.

Table F-4 Field description for link command in `/etc/llttab`

Field	Description
tag-name	A unique string that is used as a tag by LLT; for example link1, link2,....
device	The device path of the UDP protocol; for example udp. A place holder string. Linux does not have devices for protocols. So this field is ignored.
node-range	Nodes using the link. "-" indicates all cluster nodes are to be configured for this link.
link-type	Type of link; must be "rdma" for LLT over RDMA.
udp-port	Unique UDP port in the range of 49152-65535 for the link. See "Selecting UDP ports" on page 293.
MTU	"-" is the default, which has a value of 8192. Do not change this default value for the RDMA links.
IP address	IP address of the link on the local node.
bcast-address	Specify the value of the subnet broadcast address.

Selecting UDP ports

When you select a UDP port, select an available 16-bit integer from the range that follows:

- Use available ports in the private range 49152 to 65535
- Do not use the following ports:

- Ports from the range of well-known ports, 0 to 1023
- Ports from the range of registered ports, 1024 to 49151

To check which ports are defined as defaults for a node, examine the file /etc/services. You should also use the `netstat` command to list the UDP ports currently in use. For example:

```
# netstat -au | more
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address Foreign Address         State
udp        0      0 *:32768                 *:*
udp        0      0 *:956                   *:*
udp        0      0 *:tftp                  *:*
udp        0      0 *:sunrpc                *:*
udp        0      0 *:ipp                   *:*
```

Look in the UDP section of the output; the UDP ports that are listed under Local Address are already in use. If a port is listed in the /etc/services file, its associated name is displayed rather than the port number in the output.

Configuring the netmask for LLT

For nodes on different subnets, set the netmask so that the nodes can access the subnets in use. Run the following command and answer the prompt to set the netmask:

```
# ifconfig interface_name netmask netmask
```

For example:

- For the first network interface on the node sys1:

  ```
  IP address=192.168.9.1, Broadcast address=192.168.9.255,
  Netmask=255.255.255.0
  ```

  For the first network interface on the node sys2:

  ```
  IP address=192.168.9.2, Broadcast address=192.168.9.255,
  Netmask=255.255.255.0
  ```

- For the second network interface on the node sys1:

  ```
  IP address=192.168.10.1, Broadcast address=192.168.10.255,
  Netmask=255.255.255.0
  ```

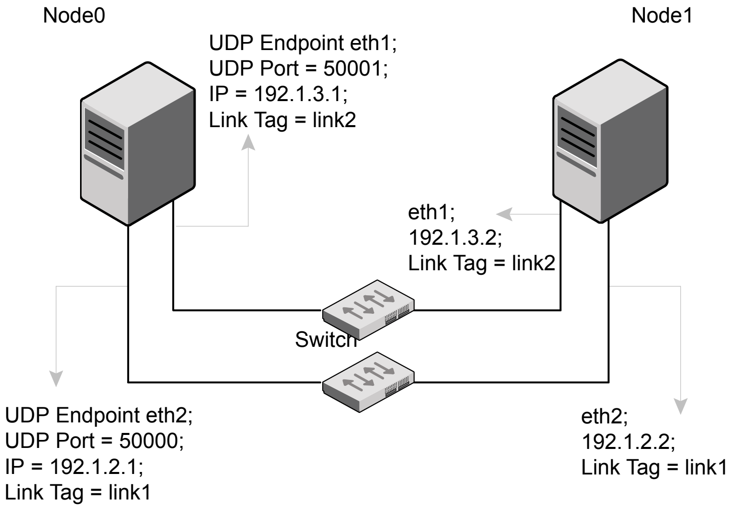
  For the second network interface on the node sys2:

```
IP address=192.168.10.2, Broadcast address=192.168.10.255,
Netmask=255.255.255.0
```

Sample configuration: direct-attached links

Figure F-1 depicts a typical configuration of direct-attached links employing LLT over UDP.

Figure F-1 A typical configuration of direct-attached links that uses LLT over RDMA



The configuration that the /etc/llttab file for Node 0 represents has directly attached crossover links. It might also have the links that are connected through a hub or switch. These links do not cross routers.

LLT sends broadcasts to peer nodes to discover their addresses. So the addresses of peer nodes do not need to be specified in the /etc/llttab file using the `set-addr` command. For direct attached links, you do need to set the broadcast address of the links in the /etc/llttab file. Verify that the IP addresses and broadcast addresses are set correctly by using the `ifconfig -a` command.

```
set-node Node0
set-cluster 1
#configure Links
#link tag-name device node-range  link-type udp port MTU IP-addressbast-address
link link1 udp - rdma 50000 - 192.1.2.1 192.1.2.255
link link2 udp - rdma 50001 - 192.1.3.1 192.1.3.255
```

The file for Node 1 resembles:

```
set-node Node1
set-cluster 1
#configure Links
#link tag-name device node-range link-type udp port MTU IP-address bast-address
link link1 udp - rdma 50000 - 192.1.2.2 192.1.2.255
link link2 udp - rdma 50001 - 192.1.3.2 192.1.3.255
```

LLT over RDMA sample /etc/llttab

The following is a sample of LLT over RDMA in the etc/llttab file.

```
set-node sys1
set-cluster clus1
link eth1 udp - rdma 50000 - 192.168.10.1 - 192.168.10.255
link eth2 udp - rdma 50001 - 192.168.11.1 - 192.168.11.255
link-lowpri eth0 udp - rdma 50004 - 10.200.58.205 - 10.200.58.255
```

Verifying LLT configuration

After starting LLT, GAB and other component, run the following commands to verify the LLT configuration.

1. Run the `lltstat -l` command to view the RDMA link configuration. View the link-type configured to rdma for the RDMA links.

```
# lltstat -l


LLT link information:
link 0  link0 on rdma hipri
mtu 8192, sap 0x2345, broadcast 192.168.27.255, addrlen 4
txpkts 171  txbytes 10492
rxpkts 105  rxbytes 5124
latehb 0  badcksum 0  errors 0
```

2. Run the `lltstat -nvv -r` command to view the RDMA and non-RDMA channel connection state.

   LLT internally configures each RDMA link in two modes (RDMA and non-RDMA) to allow both RDMA and non-RDMA traffic to use the same link. The GAB membership-related traffic goes over the non-RDMA channel while node to node data-transfer goes over high-speed RDMA channel for better performance.

```
# lltstat -rnvv active


LLT node information:
Node          State Link  Status TxRDMA RxRDMA Address
* 0 thorpc365 OPEN  link0   UP    UP     UP    192.168.27.1
                    link1   UP    UP     UP    192.168.28.1
                    link2   UP    N/A    N/A   00:15:17:97:91:2E

1 thorpc366   OPEN  link0   UP    UP     UP    192.168.27.2
                    link1   UP    UP     UP    192.168.28.2
                    link2   UP    N/A    N/A   00:15:17:97:A1:7C
```

Troubleshooting LLT over RDMA

This section lists the issues and their resolutions.

IP addresses associated to the RDMA NICs do not automatically plumb on node restart

If IP addresses do not plumb automatically, you might experience LLT failure.

Resolution: Assign unique IP addresses to RNICs and assign the same in the configuration script. For example, on an ethernet network, the `ifcfg-eth` script must be modified with the unique IP address of the RNIC.

See “Configuring IP addresses over InfiniBand Interfaces” on page 335.

Ping test fails for the IP addresses configured over InfiniBand interfaces

Resolution: Check the physical configuration and configure OpenSM. If you configured multiple links, then make sure that you have configured OpenSM to monitor multiple links in the configuration file. On RHEL and supported RHEL-compatible distributions, configure the `/etc/sysconfig/opensm` file.

See “ Configuring the OpenSM service” on page 334.

After a node restart, by default the Mellanox card with Virtual Protocol Interconnect (VPI) gets configured in InfiniBand mode

After restart, you might expect the Mellanox VPI RNIC to get configured in the Ethernet mode. By default, the card gets configured in the InfiniBand mode.

Resolution: Update the Mellanox configuration file. On RHEL and supported RHEL-compatible distributions, configure the `/etc/rdma/mlx4.conf` file.

The LLT module fails to start

For Linux distributions:

```
# systemctl start llt
```

When you try to start LLT, it may fail to start and you may see the following message:

```
Starting LLT:
LLT: loading module...
LLT:Error loading LLT dependency rdma_cm.
Make sure module rdma_cm is available on the system.
```

Description: Check the system log at /var/log/messages. If the log file lists the following error, the issue may be because the IPv6 module is not available on the system. In addition, the LLT module has indirect dependency on the IPv6 module.

```
ib_addr: Unknown symbol ipv6_dev_get_saddr
ib_addr: Unknown symbol ip6_route_output
ib_addr: Unknown symbol ipv6_chk_addr
```

Resolution: Load the IPv6 module. If you do not want to configure the IPv6 module on the node, then configure the IPv6 module to start in the disabled mode.

To start IPv6 in the disabled mode:

◆ In the `/etc/modprobe.d/` directory, create a file `ipv6.conf` and add the following line to the file

```
options ipv6 disable=1
```

The LLT module starts up without any issues once the file loads the IPv6 module in the disabled mode.

Configuring the secure shell or the remote shell for communications

This appendix includes the following topics:

About configuring secure shell or remote shell communication modes before installing products

Establishing communication between nodes is required to install Veritas InfoScale software from a remote system, or to install and configure a system. The system from which the installer is run must have permissions to run `rsh` (remote shell) or `ssh` (secure shell) utilities. You need to run the installer with superuser privileges on the systems where you plan to install the Veritas InfoScale software.

You can install products to remote systems using either secure shell (ssh) or remote shell (rsh). Veritas recommends that you use ssh as it is more secure than rsh.

You can set up ssh and rsh connections in many ways.

- You can manually set up the ssh and rsh connection with UNIX shell commands.
- You can run the `installer -comsetup` command to interactively set up ssh and rsh connection.
- You can run the password utility, `pwdutil.pl`.

This section contains an example of how to set up ssh password free communication. The example sets up ssh between a source system (sys1) that contains the installation directories, and a target system (sys2). This procedure also applies to multiple target systems.

Note: The product installer supports establishing passwordless communication.

Manually configuring passwordless ssh

The ssh program enables you to log into and execute commands on a remote system. ssh enables encrypted communications and an authentication process between two untrusted hosts over an insecure network.

In this procedure, you first create a DSA key pair. From the key pair, you append the public key from the source system to the authorized_keys file on the target systems.

Read the ssh documentation and online manual pages before enabling ssh. Contact your operating system support provider for issues regarding ssh configuration.

Visit the Openssh website that is located at: http://www.openssh.com/ to access online manuals and other resources.

To create the DSA key pair

1. On the source system (sys1), log in as root, and navigate to the root directory.

   ```
   sys1 # cd /root
   ```

2. To generate a DSA key pair on the source system, type the following command:

   ```
   sys1 # ssh-keygen -t dsa
   ```

   System output similar to the following is displayed:

   ```
   Generating public/private dsa key pair.
   Enter file in which to save the key (/root/.ssh/id_dsa):
   ```

3. Press Enter to accept the default location of `/root/.ssh/id_dsa`.

4 When the program asks you to enter the passphrase, press the Enter key twice.

```
Enter passphrase (empty for no passphrase):
```

Do not enter a passphrase. Press Enter.

```
Enter same passphrase again:
```

Press Enter again.

5 Output similar to the following lines appears.

```
Your identification has been saved in /root/.ssh/id_dsa.
Your public key has been saved in /root/.ssh/id_dsa.pub.
The key fingerprint is:
1f:00:e0:c2:9b:4e:29:b4:0b:6e:08:f8:50:de:48:d2 root@sys1
```

To append the public key from the source system to the authorized_keys file on the target system, using secure file transfer

1 From the source system (sys1), move the public key to a temporary file on the target system (sys2).

Use the secure file transfer program.

In this example, the file name `id_dsa.pub` in the root directory is the name for the temporary file for the public key.

Use the following command for secure file transfer:

```
sys1 # sftp sys2
```

If the secure file transfer is set up for the first time on this system, output similar to the following lines is displayed:

```
Connecting to sys2 ...
The authenticity of host 'sys2 (10.182.00.00)'
can't be established. DSA key fingerprint is
fb:6f:9f:61:91:9d:44:6b:87:86:ef:68:a6:fd:88:7d.
Are you sure you want to continue connecting (yes/no)?
```

2 Enter `yes`.

Output similar to the following is displayed:

```
Warning: Permanently added 'sys2,10.182.00.00'
(DSA) to the list of known hosts.
root@sys2 password:
```

3 Enter the root password of sys2.

4 At the `sftp` prompt, type the following command:

```
sftp> put /root/.ssh/id_dsa.pub
```

The following output is displayed:

```
Uploading /root/.ssh/id_dsa.pub to /root/id_dsa.pub
```

5 To quit the SFTP session, type the following command:

```
sftp> quit
```

6 Add the `id_dsa.pub` keys to the `authorized_keys` file on the target system. To begin the `ssh` session on the target system (sys2 in this example), type the following command on sys1:

```
sys1 # ssh sys2
```

Enter the root password of sys2 at the prompt:

```
password:
```

Type the following commands on sys2:

```
sys2 # cat /root/id_dsa.pub >> /root/.ssh/authorized_keys
sys2 # rm  /root/id_dsa.pub
```

7 Run the following commands on the source installation system. If your ssh session has expired or terminated, you can also run these commands to renew the session. These commands bring the private key into the shell environment and make the key globally available to the user `root`:

```
sys1 # exec /usr/bin/ssh-agent $SHELL
sys1 # ssh-add

  Identity added: /root/.ssh/id_dsa
```

This shell-specific step is valid only while the shell is active. You must execute the procedure again if you close the shell during the session.

To verify that you can connect to a target system

1. On the source system (sys1), enter the following command:

   ```
   sys1 # ssh -l root sys2 uname -a
   ```

   where sys2 is the name of the target system.

2. The command should execute from the source system (sys1) to the target system (sys2) without the system requesting a passphrase or password.
3. Repeat this procedure for each target system.

Setting up ssh and rsh connection using the installer -comsetup command

You can interactively set up the ssh and rsh connections using the `installer -comsetup` command.

Enter the following:

```
# ./installer -comsetup

Input the name of the systems to set up communication:
Enter the <platform> system names separated by spaces:
[q,?] sys2
Set up communication for the system sys2:

  Checking communication on sys2 .................. Failed

CPI ERROR V-9-20-1303 ssh permission was denied on sys2. rsh
permission was denied on sys2. Either ssh or rsh is required
to be set up and ensure that it is working properly between the local
node and sys2 for communication

Either ssh or rsh needs to be set up between the local system and
sys2 for communication

Would you like the installer to setup ssh or rsh communication
automatically between the systems?
Superuser passwords for the systems will be asked. [y,n,q,?] (y) y

Enter the superuser password for system sys2:

     1)  Setup ssh between the systems
```

```
     2)  Setup rsh between the systems
     b)  Back to previous menu

Select the communication method [1-2,b,q,?] (1) 1

Setting up communication between systems. Please wait.
Re-verifying systems.

 Checking communication on sys2 ..................... Done

Successfully set up communication for the system sys2
```

Setting up ssh and rsh connection using the pwdutil.pl utility

The password utility, `pwdutil.pl`, is bundled under the `scripts` directory. The users can run the utility in their script to set up the ssh and rsh connection automatically.

```
# ./pwdutil.pl -h
Usage:

Command syntax with simple format:

    pwdutil.pl check|configure|unconfigure ssh|rsh <hostname|IP addr>
    [<user>] [<password>] [<port>]

Command syntax with advanced format:

    pwdutil.pl [--action|-a 'check|configure|unconfigure']
               [--type|-t 'ssh|rsh']
               [--user|-u  '<user>']
               [--password|-p '<password>']
               [--port|-P '<port>']
               [--hostfile|-f '<hostfile>']
               [--keyfile|-k '<keyfile>']
               [-debug|-d]
               <host_URI>

    pwdutil.pl -h | -?
```

Table G-1 Options with pwdutil.pl utility

Option	Usage
--action|-a 'check|configure|unconfigure'	Specifies action type, default is 'check'.
--type|-t 'ssh|rsh'	Specifies connection type, default is 'ssh'.
--user|-u '<user>'	Specifies user id, default is the local user id.
--password|-p '<password>'	Specifies user password, default is the user id.
--port|-P '<port>'	Specifies port number for ssh connection, default is 22
--keyfile|-k '<keyfile>'	Specifies the private key file.
--hostfile|-f '<hostfile>'	Specifies the file which list the hosts.
-debug	Prints debug information.
-h|-?	Prints help messages.
<host_URI>	Can be in the following formats: <hostname> <user>:<password>@<hostname> <user>:<password>@<hostname>: <port>

You can check, configure, and unconfigure ssh or rsh using the `pwdutil.pl`utility. For example:

- To check ssh connection for only one host:

  ```
  pwdutil.pl check ssh hostname
  ```

- To configure ssh for only one host:

  ```
  pwdutil.pl configure ssh hostname user password
  ```

- To unconfigure rsh for only one host:

  ```
  pwdutil.pl unconfigure rsh hostname
  ```

- To configure ssh for multiple hosts with same user ID and password:

```
pwdutil.pl -a configure -t ssh -u user -p password hostname1
hostname2 hostname3
```

- To configure ssh or rsh for different hosts with different user ID and password:

  ```
  pwdutil.pl -a configure -t ssh user1:password1@hostname1
  user2:password2@hostname2
  ```

- To check or configure ssh or rsh for multiple hosts with one configuration file:

  ```
  pwdutil.pl -a configure -t ssh --hostfile /tmp/sshrsh_hostfile
  ```

- To keep the host configuration file secret, you can use the 3rd party utility to encrypt and decrypt the host file with password.
  For example:

  ```
  ### run openssl to encrypt the host file in base64 format
  # openssl aes-256-cbc -a -salt -in /hostfile -out /hostfile.enc
  enter aes-256-cbc encryption password: <password>
  Verifying - enter aes-256-cbc encryption password: <password>

  ### remove the original plain text file
  # rm /hostfile

  ### run openssl to decrypt the encrypted host file
  # pwdutil.pl -a configure -t ssh `openssl aes-256-cbc -d -a
  -in /hostfile.enc`
  enter aes-256-cbc decryption password: <password>
  ```

- To use the ssh authentication keys which are not under the default `$HOME/.ssh` directory, you can use --keyfile option to specify the ssh keys. For example:

  ```
  ### create a directory to host the key pairs:
  # mkdir /keystore

  ### generate private and public key pair under the directory:
  # ssh-keygen -t rsa -f /keystore/id_rsa

  ### setup ssh connection with the new generated key pair under
  the directory:
  # pwdutil.pl -a configure -t ssh --keyfile /keystore/id_rsa
  user:password@hostname
  ```

You can see the contents of the configuration file by using the following command:

```
# cat /tmp/sshrsh_hostfile
user1:password1@hostname1
user2:password2@hostname2
user3:password3@hostname3
user4:password4@hostname4

# all default: check ssh connection with local user
hostname5
The following exit values are returned:

0      Successful completion.
1      Command syntax error.
2      Ssh or rsh binaries do not exist.
3      Ssh or rsh service is down on the remote machine.
4      Ssh or rsh command execution is denied due to password is required.
5      Invalid password is provided.
255    Other unknown error.
```

Restarting the ssh session

After you complete this procedure, ssh can be restarted in any of the following scenarios:

- After a terminal session is closed
- After a new terminal session is opened
- After a system is restarted
- After too much time has elapsed, to refresh ssh

To restart ssh

1 On the source installation system (sys1), bring the private key into the shell environment.

```
sys1 # exec /usr/bin/ssh-agent $SHELL
```

2 Make the key globally available for the user `root`

```
sys1 # ssh-add
```

Enabling rsh for Linux

The following section describes how to enable remote shell.

Veritas recommends configuring a secure shell environment for Veritas InfoScale product installations.

See “Manually configuring passwordless ssh” on page 347.

See the operating system documentation for more information on configuring remote shell.

To enable rsh for RHEL

- Run the following commands to enable rsh passwordless connection:

  ```
  # systemctl start rsh.socket
  # systemctl start rlogin.socket
  # systemctl enable rsh.socket
  # systemctl enable rlogin.socket
  # echo rsh >> /etc/securetty
  # echo rlogin >> /etc/securetty
  #echo “+ +” >> /root/.rhosts
  ```

To disable rsh for RHEL

- Run the following commands to disable rsh passwordless connection:

  ```
  # systemctl stop rsh.socket
  # systemctl stop rlogin.socket
  # systemctl disable rsh.socket
  # systemctl disable rlogin.socket
  ```

Installation script options

This appendix includes the following topics:

- Installation script options

Installation script options

Table H-1 shows command line options for the installation script. For an initial install or upgrade, options are not usually required. The installation script options apply to all Veritas InfoScale product scripts, except where otherwise noted.

Table H-1 Available command line options

Command Line Option	Function
-addnode	Adds a node to a high availability cluster.
-allpkgs	Displays all RPMs required for the specified product. The RPMs are listed in correct installation order. The output can be used to create scripts for command line installs, or for installations over a network.
-comcleanup	The `-comcleanup` option removes the secure shell or remote shell configuration added by installer on the systems. The option is only required when installation routines that performed auto-configuration of the shell are abruptly terminated.
-comsetup	The `-comsetup` option is used to set up the ssh or rsh communication between systems without requests for passwords or passphrases.

Table H-1 Available command line options *(continued)*

Command Line Option	Function
-configcps	The `-configcps` option is used to configure CP server on a running system or cluster.
-configure	Configures the product after installation.
-disable_dmp_native_support	Disables Dynamic Multi-pathing support for the native LVM volume groups and ZFS pools during upgrade. Retaining Dynamic Multi-pathing support for the native LVM volume groups and ZFS pools during upgrade increases RPM upgrade time depending on the number of LUNs and native LVM volume groups and ZFS pools configured on the system.
-fencing	Configures I/O fencing in a running cluster.
-fips	The `-fips` option is used to enable or disable security with fips mode on a running VCS cluster. It could only be used together with `-security` or `-securityonenode` option.
–hostfile *full_path_to_file*	Specifies the location of a file that contains a list of hostnames on which to install.
-install	Used to install products on system
-online_upgrade	Used to perform online upgrade. Using this option, the installer upgrades the whole cluster and also supports customer's application zero down time during the upgrade procedure. Now this option only supports VCS and ApplicationHA.
-patch_path	Defines the path of a patch level release to be integrated with a base or a maintenance level release in order for multiple releases to be simultaneously installed .
-patch2_path	Defines the path of a second patch level release to be integrated with a base or a maintenance level release in order for multiple releases to be simultaneously installed.

Table H-1 Available command line options *(continued)*

Command Line Option	Function
-patch3_path	Defines the path of a third patch level release to be integrated with a base or a maintenance level release in order for multiple releases to be simultaneously installed.
-patch4_path	Defines the path of a fourth patch level release to be integrated with a base or a maintenance level release in order for multiple releases to be simultaneously installed.
-patch5_path	Defines the path of a fifth patch level release to be integrated with a base or a maintenance level release in order for multiple releases to be simultaneously installed.
–keyfile *ssh_key_file*	Specifies a key file for secure shell (SSH) installs. This option passes `–I ssh_key_file` to every SSH invocation.
–kickstart *dir_path*	Produces a kickstart configuration file for installing with Linux RHEL Kickstart. The file contains the list of required RPMs in the correct order for installing, in a format that can be used for Kickstart installations. The *dir_path* indicates the path to the directory in which to create the file.
-license	Registers or updates product licenses on the specified systems.
–logpath *log_path*	Specifies a directory other than `/opt/VRTS/install/logs` as the location where installer log files, summary files, and response files are saved.
-noipc	Disables the installer from making outbound networking calls to Veritas Services and Operations Readiness Tool (SORT) in order to automatically obtain patch and release information updates.
-nolic	Allows installation of product RPMs without entering a license key. Licensed features cannot be configured, started, or used when this option is specified.

Table H-1 Available command line options *(continued)*

Command Line Option	Function
-pkgtable	Displays product's RPMs in correct installation order by group.
–postcheck	Checks for different HA and file system-related processes, the availability of different ports, and the availability of cluster-related service groups.
-precheck	Performs a preinstallation check to determine if systems meet all installation requirements. Veritas recommends doing a precheck before installing a product.
-prod	Specifies the product for operations.
-component	Specifies the component for operations.
-redirect	Displays progress details without showing the progress bar.
-require	Specifies an installer patch file.
–responsefile *response_file*	Automates installation and configuration by using system and configuration information stored in a specified file instead of prompting for information. The *response_file* must be a full path name. You must edit the response file to use it for subsequent installations. Variable field definitions are defined within the file.
-rsh	Specify this option when you want to use RSH and RCP for communication between systems instead of the default SSH and SCP. See "About configuring secure shell or remote shell communication modes before installing products" on page 346.
-security	The -security option is used to convert a running VCS cluster between secure and non-secure modes of operation.
-securityonenode	The `-securityonenode` option is used to configure a secure cluster node by node.
-securitytrust	The `-securitytrust` option is used to setup trust with another broker.

Table H-1 Available command line options *(continued)*

Command Line Option	Function
–serial	Specifies that the installation script performs install, uninstall, start, and stop operations on each system in a serial fashion. If this option is not specified, these operations are performed simultaneously on all systems.
-settunables	Specify this option when you want to set tunable parameters after you install and configure a product. You may need to restart processes of the product for the tunable parameter values to take effect. You must use this option together with the `-tunablesfile` option.
-start	Starts the daemons and processes for the specified product.
-stop	Stops the daemons and processes for the specified product.
-timeout	The `-timeout` option is used to specify the number of seconds that the script should wait for each command to complete before timing out. Setting the `-timeout` option overrides the default value of 1200 seconds. Setting the `-timeout` option to 0 prevents the script from timing out. The `-timeout` option does not work with the `-serial option`
–tmppath *tmp_path*	Specifies a directory other than `/opt/VRTStmp` as the working directory for the installation scripts. This destination is where initial logging is performed and where RPMs are copied on remote systems before installation.
-tunables	Lists all supported tunables and create a tunables file template.
-tunables_file *tunables_file*	Specify this option when you specify a tunables file. The tunables file should include tunable parameters.
-uninstall	This option is used to uninstall the products from systems

Table H-1 Available command line options *(continued)*

Command Line Option	Function
-upgrade	Specifies that an existing version of the product exists and you plan to upgrade it.
-version	Checks and reports the installed products and their versions. Identifies the installed and missing RPMs and patches where applicable for the product. Provides a summary that includes the count of the installed and any missing RPMs and patches where applicable. Lists the installed patches, patches, and available updates for the installed product if an Internet connection is available.
-yumgroupxml	The `-yumgroupxml` option is used to generate a yum group definition XML file. The `createrepo` command can use the file on Redhat Linux to create a yum group for automated installation of all RPMs for a product. An available location to store the XML file should be specified as a complete path. The `-yumgroupxml` option is supported on Redhat Linux and supported RHEL compatible distributions only.

Troubleshooting VCS configuration

This appendix includes the following topics:

Restarting the installer after a failed network connection

If an installation is aborted because of a failed network connection, restarting the installer will detect the previous installation. The installer prompts to resume the installation. If you choose to resume the installation, the installer proceeds from the point where the installation aborted. If you choose not to resume, the installation starts from the beginning.

Cannot launch the cluster view link

If you launch the cluster view using the URL, https://<hostname_or_ip>:5634/vcs/admin/application_health.html link, the system responds with the HTTP 404 NOT FOUND error. This is possible if the xprtld configuration has not been set properly. You need to correct the configuration on each cluster node.

Run the following commands on each cluster node.

```
# echo 'namespaces vcs=/opt/VRTSvcs/portal/vcs_wizards' >>
/etc/opt/VRTSsfmh/xprtld.conf

# /opt/VRTSsfmh/bin/xprtlc -l \

https://localhost/admin/xprtld/config/namespace/add \

-d namespace=vcs -d document_root="/opt/VRTSvcs/portal/vcs_wizards"
```

Relaunch the cluster view link.

Starting and stopping processes for the Veritas InfoScale products

After the installation and configuration is complete, the Veritas InfoScale product installer starts the processes that the installed products use. You can use the product installer to stop or start the processes, if required.

To stop the processes

◆ Use the `-stop` option to stop the product installation script.

  For example, to stop the product's processes, enter the following command:

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

To start the processes

◆ Use the `-start` option to start the product installation script.

  For example, to start the product's processes, enter the following command:

  ```
  # ./installer -start
  ```

Installer cannot create UUID for the cluster

The installer displays the following error message if the installer cannot find the uuidconfig.pl script before it configures the UUID for the cluster:

```
Couldn't find uuidconfig.pl for uuid configuration,
please create uuid manually before start vcs
```

You may see the error message during VCS configuration, upgrade, or when you add a node to the cluster using the installer.

Workaround: To start VCS, you must run the uuidconfig.pl script manually to configure the UUID on each cluster node.

To configure the cluster UUID when you create a cluster manually

- On one node in the cluster, perform the following command to populate the cluster UUID on each node in the cluster.

  ```
  # /opt/VRTSvcs/bin/uuidconfig.pl -clus -configure nodeA
  nodeB ... nodeN
  ```

  Where nodeA, nodeB, through nodeN are the names of the cluster nodes.

LLT startup script displays errors

If more than one system on the network has the same clusterid-nodeid pair and the same Ethernet sap/UDP port, then the LLT startup script displays error messages similar to the following:

```
LLT lltconfig ERROR V-14-2-15238 node 1 already exists
in cluster 8383 and has the address - 00:18:8B:E4:DE:27
LLT lltconfig ERROR V-14-2-15241 LLT not configured,
use -o to override this warning
LLT lltconfig ERROR V-14-2-15664 LLT could not
configure any link
LLT lltconfig ERROR V-14-2-15245 cluster id 1 is
already being used by nid 0 and has the
address - 00:04:23:AC:24:2D
LLT lltconfig ERROR V-14-2-15664 LLT could not
configure any link
```

Recommended action: Ensure that all systems on the network have unique clusterid-nodeid pair. You can use the `lltdump -f` *`device`* `-D` command to get the list of unique clusterid-nodeid pairs connected to the network. This utility is available only for LLT-over-ethernet.

The vxfentsthdw utility fails when SCSI TEST UNIT READY command fails

While running the vxfentsthdw utility, you may see a message that resembles as follows:

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

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

The GAB program reports incorrect membership results with existing iptable rules

With an LLT over UDP or LLT over RDMA configuration, when you start a cluster, GAB may not show membership details even though LLT links are set up correctly. It could be because some iptable rules filter out LLT packets.

Resolution: Add iptable rules to unblock LLT packets:

```
# iptables -I INPUT -i interface-name -p udp --dport
UDP_portnumber_LLT -j ACCEPT
```

```
# iptables -I INPUT -i interface-name -p udp --sport
UDP_portnumber_LLT -j ACCEPT
```

For example, if the interfaces eth2 and eth3 are configured under LLT with port numbers 50000 and 50001 respectively, the commands are:

```
# iptables -I INPUT -i eth2 -p udp --dport 50000 -j ACCEPT
```

```
# iptables -I INPUT -i eth2 -p udp --sport 50000 -j ACCEPT
```

```
# iptables -I INPUT -i eth3 -p udp --dport 50001 -j ACCEPT
```

```
# iptables -I INPUT -i eth3 -p udp --sport 50001 -j ACCEPT
```

Append these rules to the `/etc/sysconfig/iptables` file to make these changes persistent:

```
-A INPUT -p udp --dport UDP_portnumber1_LLT -j ACCEPT
```

```
-A INPUT -p udp --sport UDP_portnumber1_LLT -j ACCEPT

-A INPUT -p udp --dport UDP_portnumber2_LLT -j ACCEPT
-A INPUT -p udp --sport UDP_portnumber2_LLT -j ACCEPT


For example,
-A INPUT -p udp --dport 50001 -j ACCEPT
-A INPUT -p udp --sport 50001 -j ACCEPT

-A INPUT -p udp --dport 50002 -j ACCEPT
-A INPUT -p udp --sport 50002 -j ACCEPT
```

Issues during fencing startup on VCS cluster nodes set up for server-based fencing

Table I-1 Fencing startup issues on VCS cluster (client cluster) nodes

Issue	Description and resolution
`cpsadm` command on the VCS cluster gives connection error	If you receive a connection error message after issuing the `cpsadm` command on the VCS cluster, perform the following actions: ■ Ensure that the CP server is reachable from all the VCS cluster nodes. ■ Check the /etc/vxfenmode file and ensure that the VCS cluster nodes use the correct CP server virtual IP or virtual hostname and the correct port number. ■ For HTTPS communication, ensure that the virtual IP and ports listed for the server can listen to HTTPS requests.
Authorization failure	Authorization failure occurs when the nodes on the client clusters and or users are not added in the CP server configuration. Therefore, fencing on the VCS cluster (client cluster) node is not allowed to access the CP server and register itself on the CP server. Fencing fails to come up if it fails to register with a majority of the coordination points. To resolve this issue, add the client cluster node and user in the CP server configuration and restart fencing. See “Preparing the CP servers manually for use by the VCS cluster” on page 163.

Table I-1 Fencing startup issues on VCS cluster (client cluster) nodes *(continued)*

Issue	Description and resolution
Authentication failure	If you had configured secure communication between the CP server and the VCS cluster (client cluster) nodes, authentication failure can occur due to the following causes: ■ The client cluster requires its own private key, a signed certificate, and a Certification Authority's (CA) certificate to establish secure communication with the CP server. If any of the files are missing or corrupt, communication fails. ■ If the client cluster certificate does not correspond to the client's private key, communication fails. ■ If the CP server and client cluster do not have a common CA in their certificate chain of trust, then communication fails.

Sample VCS cluster setup diagrams for CP server-based I/O fencing

This appendix includes the following topics:

- Configuration diagrams for setting up server-based I/O fencing

Configuration diagrams for setting up server-based I/O fencing

The following CP server configuration diagrams can be used as guides when setting up CP server within your configuration:

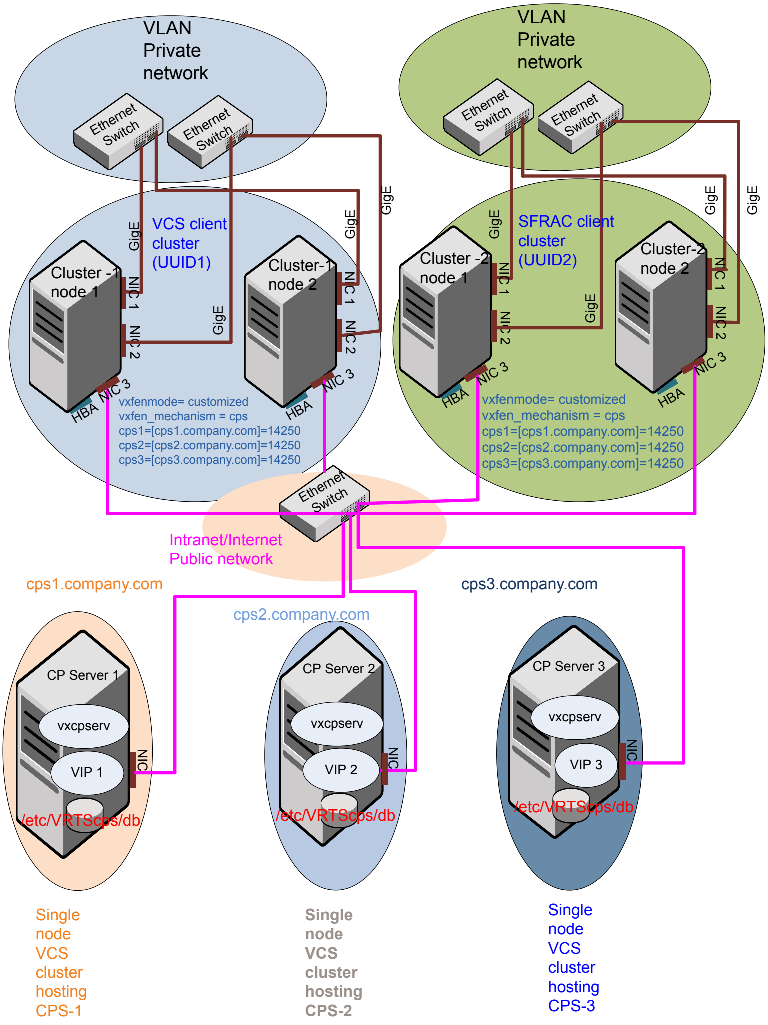
- Two unique client clusters that are served by 3 CP servers:
  See Figure J-1 on page 369.
- Client cluster that is served by highly available CP server and 2 SCSI-3 disks:
- Two node campus cluster that is served be remote CP server and 2 SCSI-3 disks:
- Multiple client clusters that are served by highly available CP server and 2 SCSI-3 disks:

Two unique client clusters served by 3 CP servers

Figure J-1 displays a configuration where two unique client clusters are being served by 3 CP servers (coordination points). Each client cluster has its own unique user ID (UUID1 and UUID2).

In the `vxfenmode` file on the client nodes, vxfenmode is set to `customized` with vxfen mechanism set to `cps`.

Figure J-1 Two unique client clusters served by 3 CP servers



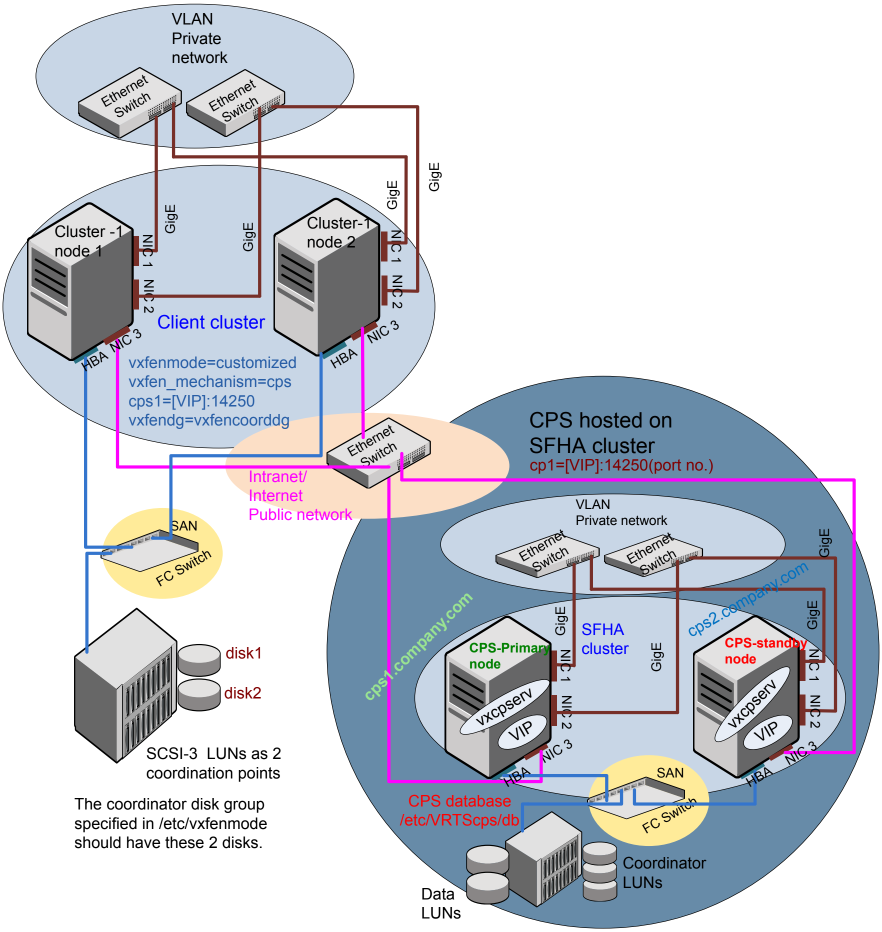
Client cluster served by highly available CPS and 2 SCSI-3 disks

Figure J-2 displays a configuration where a client cluster is served by one highly available CP server and 2 local SCSI-3 LUNs (disks).

In the `vxfenmode` file on the client nodes, `vxfenmode` is set to customized with vxfen mechanism set to `cps`.

The two SCSI-3 disks are part of the disk group vxfencoorddg. The third coordination point is a CP server hosted on an SFHA cluster, with its own shared database and coordinator disks.

Figure J-2 Client cluster served by highly available CP server and 2 SCSI-3 disks



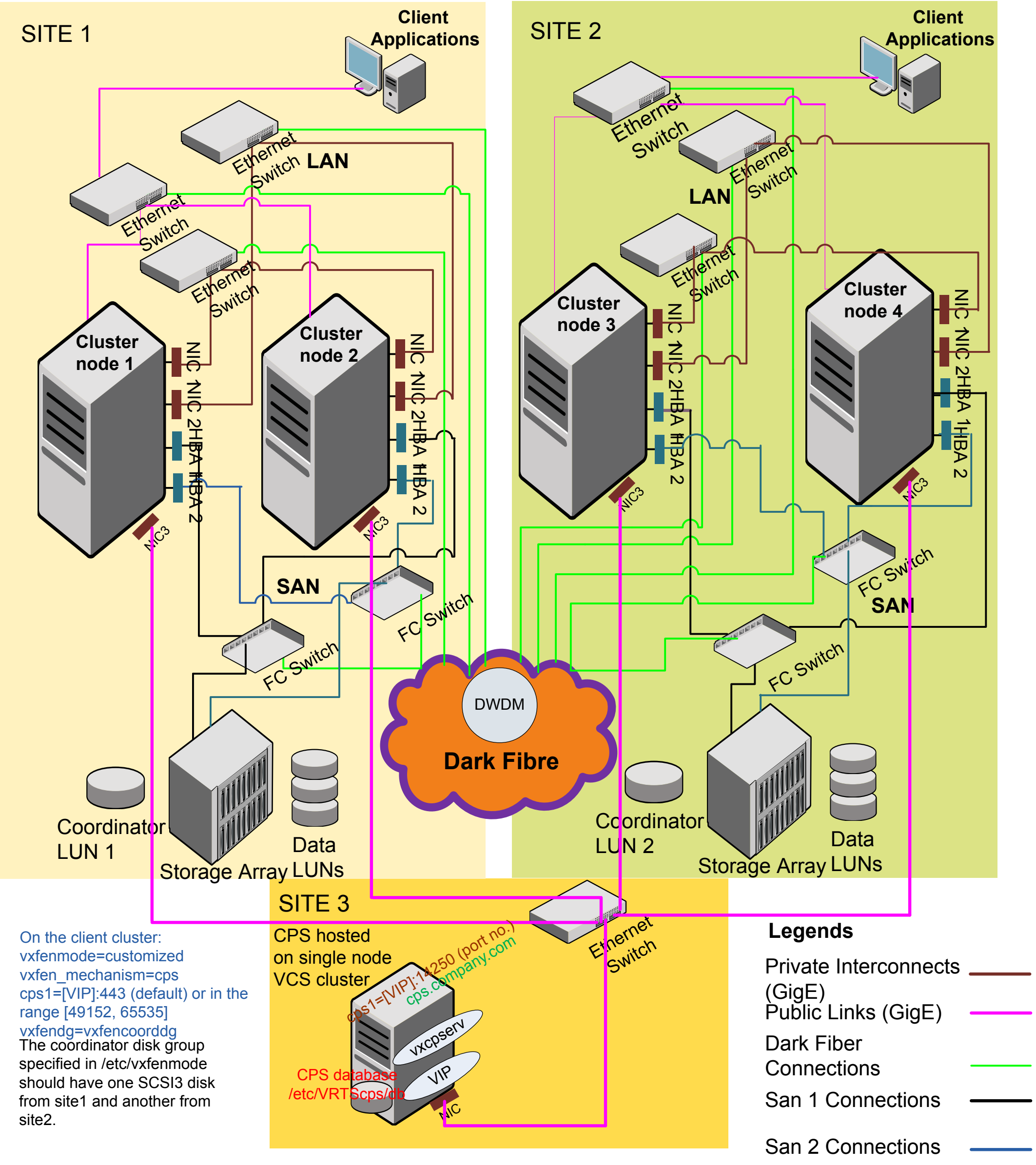
Two node campus cluster served by remote CP server and 2 SCSI-3 disks

Figure J-3 displays a configuration where a two node campus cluster is being served by one remote CP server and 2 local SCSI-3 LUN (disks).

In the `vxfenmode` file on the client nodes, vxfenmode is set to `customized` with vxfen mechanism set to `cps`.

The two SCSI-3 disks (one from each site) are part of disk group vxfencoorddg. The third coordination point is a CP server on a single node VCS cluster.

Figure J-3 Two node campus cluster served by remote CP server and 2 SCSI-3



On the client cluster:
vxfenmode=customized
vxfen_mechanism=cps
cps1=[VIP]:443 (default) or in the range [49152, 65535]
vxfendg=vxfencoorddg
The coordinator disk group specified in /etc/vxfenmode should have one SCSI3 disk from site1 and another from site2.

Legends

Private Interconnects (GigE) ━━

Public Links (GigE) ━━

Dark Fiber Connections ━━

San 1 Connections ━━

San 2 Connections ━━

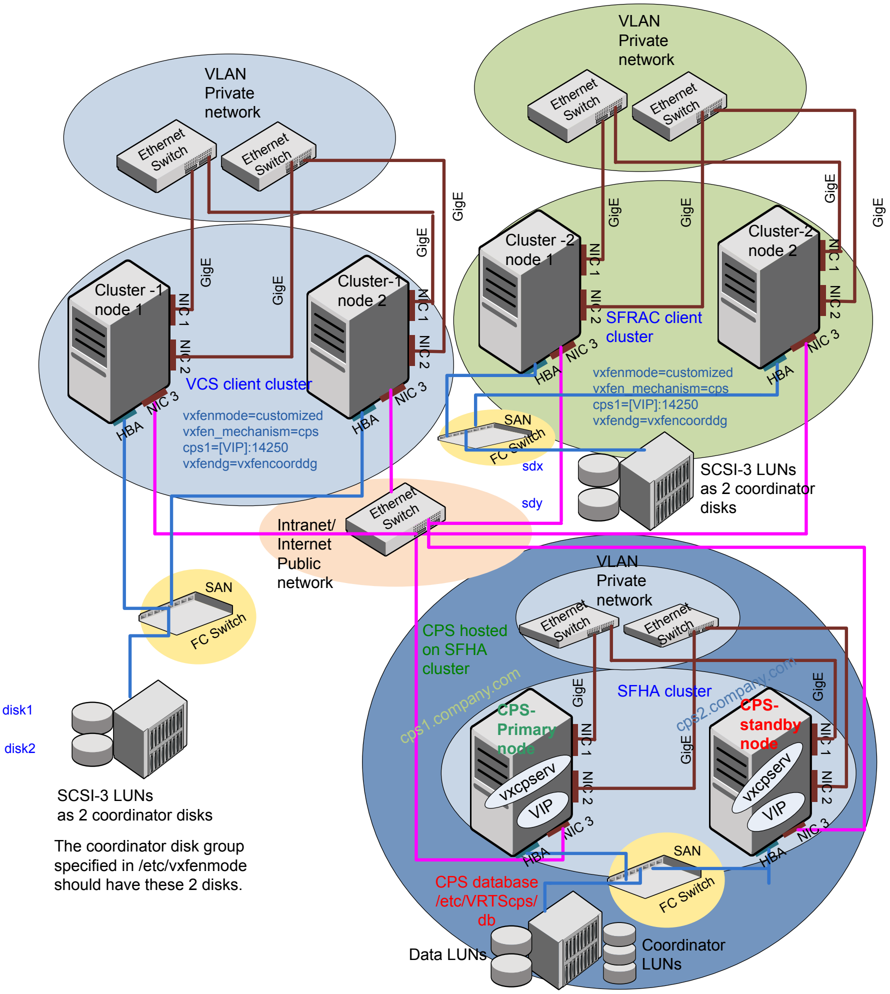
Multiple client clusters served by highly available CP server and 2 SCSI-3 disks

Figure J-4 displays a configuration where multiple client clusters are being served by one highly available CP server and 2 local SCSI-3 LUNS (disks).

In the `vxfenmode` file on the client nodes, vxfenmode is set to `customized` with vxfen mechanism set to `cps`.

The two SCSI-3 disks are are part of the disk group vxfencoorddg. The third coordination point is a CP server, hosted on an SFHA cluster, with its own shared database and coordinator disks.

Figure J-4 Multiple client clusters served by highly available CP server and 2 SCSI-3 disks



vxfenmode=customized
vxfen_mechanism=cps
cps1=[VIP]:14250
vxfendg=vxfencoorddg

The coordinator disk group specified in /etc/vxfenmode should have these 2 disks.

Appendix K

Upgrading the Steward process

This appendix includes the following topics:

- Upgrading the Steward process

Upgrading the Steward process

The Steward process can be configured in both secure and non-secure mode. The following procedures provide the steps to upgrade the Steward process.

Upgrading Steward configured in secure mode to 7.0

To upgrade Steward configured in secure mode:

1 Log on to the Steward system as a root user.

2 Stop the Steward process.

```
# steward -stop -secure
```

3 Upgrade the VRTSvcs and VRTSperl RPMs using:

```
# rpm -Uvh
```

4 Start the Steward process.

```
# steward -start -secure
```

Upgrading Steward configured in non-secure mode to 7.0

To upgrade Steward configured in non-secure mode:

1. Log on to the Steward system as a root user.
2. Stop the Steward process.

   ```
   # steward -stop
   ```

3. Copy and replace the Steward binary from a node in the VCS cluster to the Steward system. The file resides in the `/opt/VRTSvcs/bin/` directory. Make sure that the source from where you copy the binary is also running the same version of Linux as the destination system.
4. Start the Steward process.

   ```
   # steward -start
   ```

Refer to *About the Steward process: Split-brain in two-cluster global clusters* in the *Cluster Server Administrator's Guide* for more information.