VERITAS Cluster Server™ 1.3.0

Installation Guide

Solaris

October 2000
30-000036-399

VERITAS

Disclaimer

The information contained in this publication is subject to change without notice. VERITAS Software Corporation makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. VERITAS Software Corporation shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this manual.

Copyright

Printed in the USA, October 2000.

VERITAS Software Corporation.
1600 Plymouth St.
Mountain View, CA 94043
Phone 650.335.8000
Fax 650.335.8050
http://www.veritas.com

Contents

Preface

This guide provides information on how to install VERITAS® Cluster Server™ (VCS) on Solaris. It is intended for system and network administrators responsible for installing and configuring VCS.

- For information on using and configuring VCS, see the *VERITAS Cluster Server User's Guide*.
- For information on using VCS bundled agents, see the *VCS Bundled Agents Reference Guide.*
- For more information on the API provided by the VCS agent framework, and for instructions on how to build and test an agent, see the *VERITAS Cluster Server Agent Developer's Guide*.

How This Guide Is Organized

Chapter 1, "Introduction," describes VCS briefly; for a more comprehensive description of VCS, see the *VERITAS Cluster Server User's Guide.*

Chapter 2, "Installing VCS," instructs you on how to install and configure your hardware, and how to install VCS to all systems in a cluster via a script running on a single system. It describes how to verify the installation.

Chapter 3, "Installing VCS Using pkgadd" describes an alternate method of installing VCS in the cluster one system at a time.

Technical Support

For assistance with this VERITAS product, or for information regarding VERITAS service packages, contact Technical Support at 800.342.0652 (U.S. and Canada). You may also contact Technical Support via email at support@veritas.com.

For Customers Outside U.S. and Canada

From Europe, the Middle East, or Asia, visit the Technical Support website at http://support.veritas.com for a list of each country's contact information.

Conventions

Typeface	Usage
`courier`	computer output, files, attribute names, device names, and directories
`courier` (bold)	user input and commands, keywords in grammar syntax
italic	new terms, titles, emphasis
italic	variables within a command
Symbol	**Usage**
%	C shell prompt
$	Bourne/Korn shell prompt
#	Superuser prompt (for all shells)

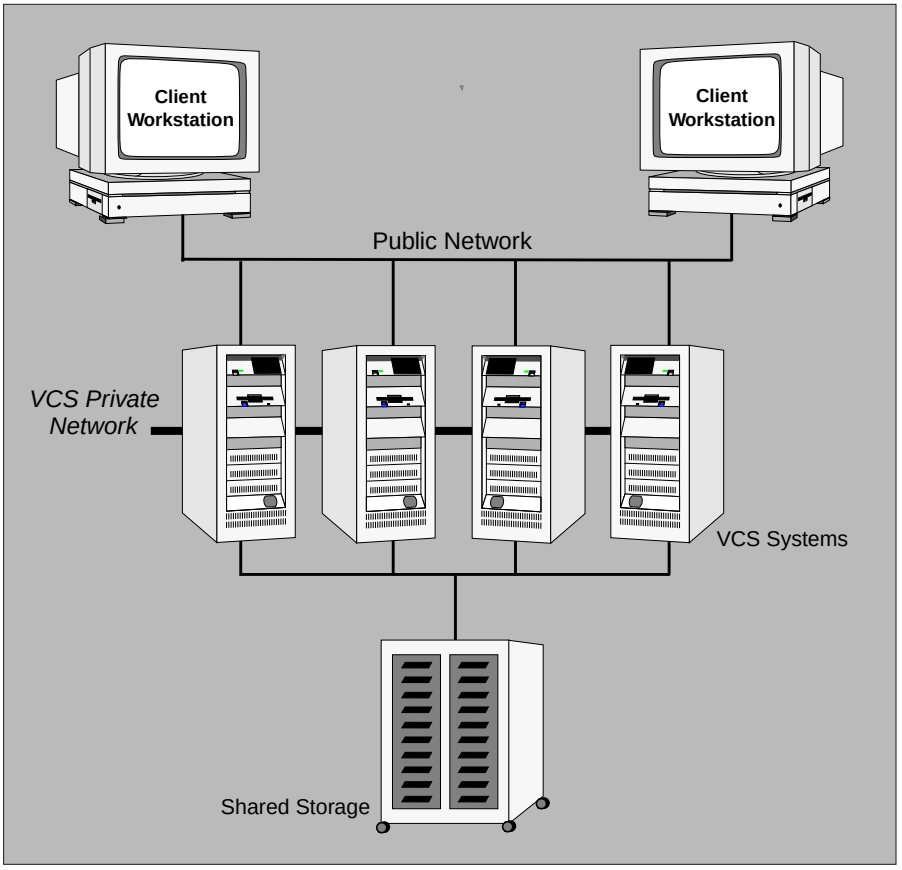
Introduction 1

VERITAS® Cluster Server™ (VCS) is a high-availability solution for cluster configurations. VCS enables you to monitor systems and application services, and to restart services on a different system when hardware or software fails.

VCS Basics

A single VCS cluster consists of multiple systems connected in various combinations to shared storage devices. VCS monitors and controls applications running in the cluster, and restarts applications in response to a variety of hardware or software faults. Client applications continue operation with little or no downtime. In some cases, such as NFS, this continuation is transparent to high-level applications and to users. In other cases, the operation must be retried; for example, a Web page must be reloaded.

The illustration on page 2 shows a typical VCS configuration of four systems connected to shared storage. Client workstations receive service over the public network from applications running on the VCS systems. VCS monitors the systems and their services. VCS systems in the cluster communicate over a private network.



Example of a Four-System VCS Cluster

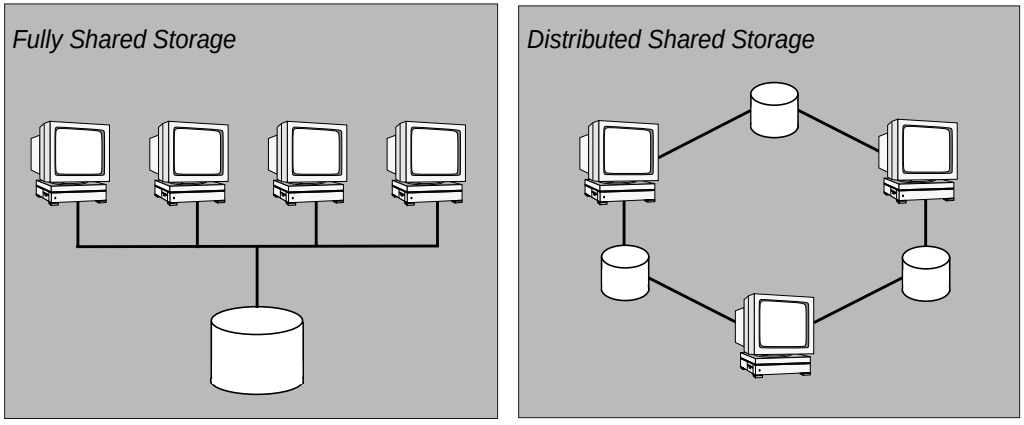
Multiple Systems

VCS runs in a replicated state on each system in the cluster. A private network enables the systems to share identical state information about all resources and to recognize which systems are active, which are joining or leaving the cluster, and which have failed. For the private network, two communication channels are required to guard against network partitions.

Shared Storage

A VCS hardware configuration typically consists of multiple systems connected to shared storage via I/O channels. Shared storage provides multiple systems an access path to the same data, and enables VCS to restart applications on alternate systems when a system fails, thus ensuring high availability.

The figures below illustrate the flexibility of VCS shared storage configurations. (Note that VCS systems can only access storage that is physically attached.)



Two Examples of Shared Storage Configurations

LLT and GAB

VCS uses two components, LLT and GAB, to share data over private networks among systems. These components provide the performance and reliability required by VCS.

- LLT (Low Latency Transport) provides fast, kernel-to-kernel communications, and monitors network connections. The system administrator configures LLT by creating a configuration file (`lltab`) that describes the systems in the cluster and the private network links among them.
- GAB (Group Membership and Atomic Broadcast) provides the global message order required to maintain a synchronized state among the systems, and monitors disk communications such as that required by the VCS heartbeat utility. The system administrator configures GAB driver by creating a configuration file (`gabtab`).

See "LLT and GAB Configuration Files" on page 23.

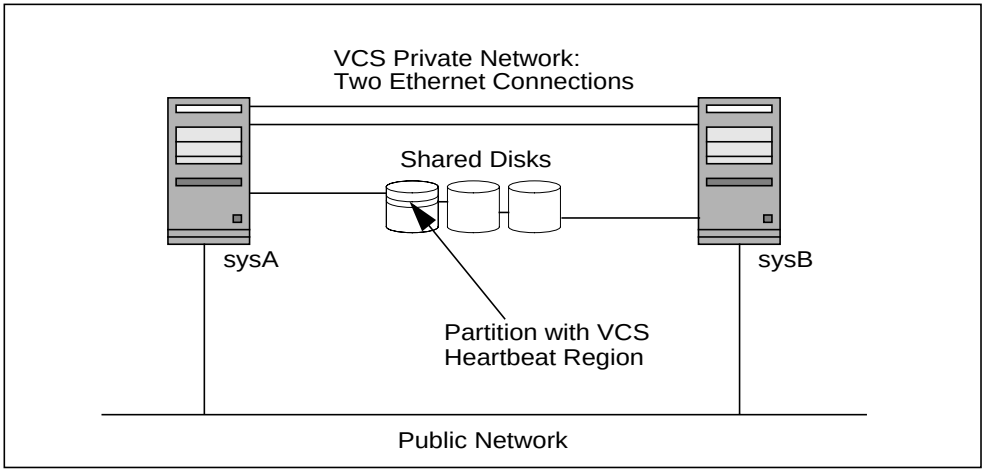
Two Types of Channels: Network and Shared Disks

For the VCS private network, there are two types of channels available for heartbeating: network connections and heartbeat regions on shared disks. The shared disk region heartbeat channel is used for heartbeating only, not for transmitting information as are network channels. For information on configuring heartbeat regions on shared disks, see "Configuring Membership Heartbeat Regions on Disk" on page 32.

Each cluster configuration requires at least two channels between systems, one of which *must* be a network connection. The remaining channels may be a combination of network connections and heartbeat regions on shared disks.

This requirement for two channels protects your cluster against network partitioning. (For more about network partitioning, refer to the *VERITAS Cluster Server User's Guide*.) We recommend configuring at least one heartbeat disk region on each I/O chain shared between systems in addition to private network connections. VCS supports a maximum of eight network channels and four heartbeat disks. For configurations using heartbeat disk regions, the cluster size is currently limited to eight systems.

The following illustration shows a two-system VCS cluster in which sysA and sysB have two private network connections and another connection via the heartbeat disk region on one of the shared disks. If one of the network connections fails, two channels remain. If both network connections fail, the condition is in jeopardy, but connectivity remains via the heartbeat disk.



Two Systems Connected by Two Ethernet Connections and a Heartbeat Disk Region

Preexisting Network Partitions

A *preexisting network partition* refers to a failure in communication channels that occurs while the systems are down and VCS cannot respond. When the systems are booted, VCS is vulnerable to network partitioning, regardless of the cause to the failure.

VCS Seeding

To protect your cluster from a preexisting network partition, VCS employs the concept of a *seed*. By default, when a system comes up, it is not *seeded*. Systems can be seeded automatically or manually. Note that only systems that have been seeded can run VCS.

Systems are seeded automatically in one of two ways:

- When an unseeded system communicates with a seeded system.
- When all systems in the cluster are unseeded and able to communicate with each other.

VCS requires that you declare the number of systems that will participate in the cluster. When the last system is booted, the cluster will seed and start VCS on all systems. Systems can then be brought down and restarted in any combination. Seeding is automatic as long as at least one instance of VCS is running somewhere in the cluster. Manual seeding is required only to run VCS from a cold start (all systems down) when not all systems are available.

Installing VCS 2

VCS Installation Tasks

Installing VCS involves various tasks, and each task must be performed in the order presented below. Detailed instructions begin on page 8.

- ✔ Set up the hardware (page 8).
- ✔ Install VCS using `InstallVCS` (page 14).
- ✔ Verify the installation (page 23).
- ✔ Verify LLT, GAB, and Cluster operation (page 25).
- ✔ Configure membership heartbeat disk regions for GAB and VCS (page 32).
- ✔ Initialize file systems and disk groups on shared storage (page 35).
- ✔ Prepare for NFS services (page 38).

After completing these procedures, you are ready to configure VCS. Refer to the *VERITAS Cluster Server User's Guide* for more information on how to configure VCS.

Setting Up the Network and Storage for a VCS Cluster

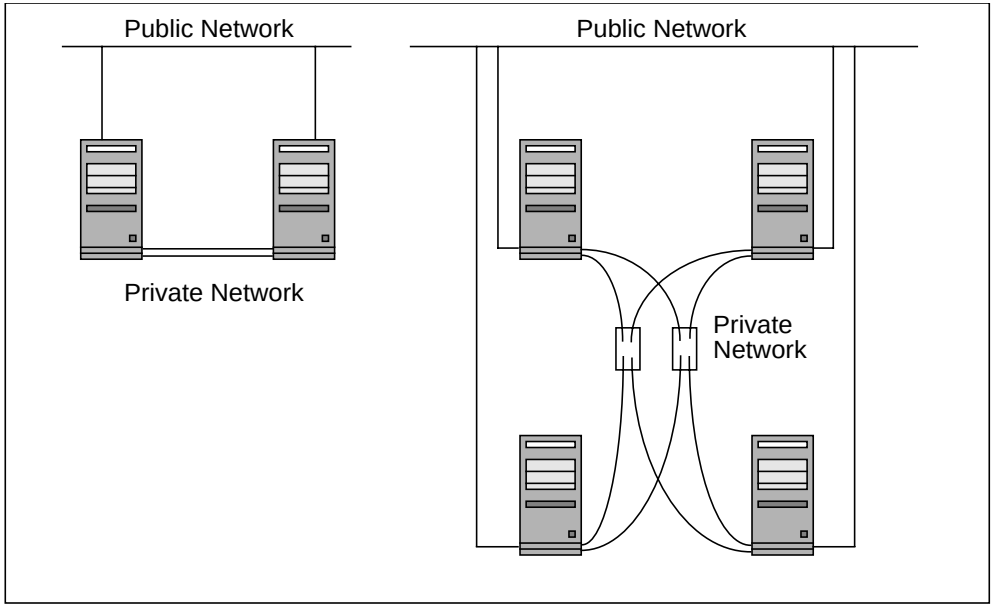
A VCS cluster requires the following hardware:

Item	Description
VCS systems	SPARC systems running Solaris 2.5.1 or later.
CD-ROM drive	One CD-ROM drive on each system, or a drive accessible to each.
Disks	Typical VCS configurations require shared disks to support applications that migrate between systems in the cluster.
Disk space	Each VCS system must have at least 35 megabytes of free space in the `/opt` file system.
Ethernet controllers	In addition to the built-in public Ethernet controller, VCS requires at least one more Ethernet interface per system. Two additional are recommended.
SCSI adapters	VCS requires at least one built-in SCSI adapter per system for the operating system disks, and at least one additional SCSI adapter per system for shared data disks.
RAM	Each VCS system requires at least 128 megabytes.

Setting Up the Network

1. Install the required Ethernet network interface cards.

2. Connect the VCS private Ethernet controllers on each system. Use cross-over Ethernet cables (supported only on two systems), or independent hubs, for each VCS communication network.



Private Network Setups: Two-Node Cluster and Four-Node Cluster

Setting Up Shared Storage

1. Install the required SCSI controllers.

2. To probe your shared SCSI buses and select IDs for the systems, cable the shared devices to one system and terminate the SCSI bus.

   a. From the `EEPROM` prompt (`ok`) type the following commands:

   ```
   ok show-devs
   ok probe-scsi-all
   ```

   b. Select a unique SCSI ID for each system on the shared SCSI bus. The priority of SCSI IDs is 7 to 0, followed by 15 to 8. Use high-priority IDs for the systems, and low-priority IDs for devices such as disks and tape drives. For example, use 7, 6, 5, and so forth, for the systems, and use the remaining IDs for devices.

3. Modify the configuration file of the shared SCSI driver on each system.

   a. Identify the three-letter name and device parent for the shared SCSI controller. Type the following command on the system connected to the shared devices:

   ```
   # ls -l shared_disk_partition
   ```

   The variable *shared_disk_partition* refers to the device path for any device on the shared SCSI bus.

   For example, type:

   ```
   # ls -l /dev/dsk/c1t0d0s3
   ```

   Output resembles:

   ```
   lrwxrwxrwx 1 root root 53 Dec 03 11:10
         /dev/dsk/c1t0d0s3 -> ../../
         devices/sbus@1f,0/QLGC,isp@0,10000/sd@0,0:d,raw
   ```

   Common SCSI driver names include `isp` (in bold text in the preceding sample output) and `fas` (not shown).

   Note that the parent name `/sbus@1f,0` (also in bold text in the preceding sample output) includes the slash after the word `devices`, and extends to, but does not include, the slash preceding the driver name.

b. Identify the register property values for the shared SCSI controller. Type the following command on the system connected to the shared devices:

```
# prtconf -v
```

Output resembles:

```
QLGC,isp, instance #0
...
Register Specifications:
   Bus Type=0x0, Address=0x10000, Size=1c2
...
```

c. Modify `/kernel/drv/`*driver_name*`.conf` on each system to set the SCSI ID for the system to use on the shared bus. (Create this file if it does not exist.)

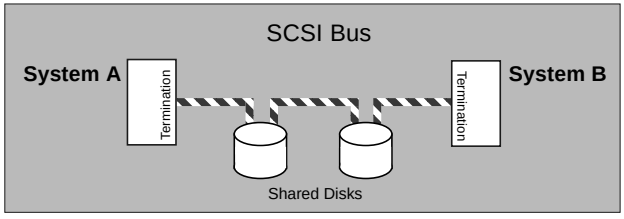
```
name="driver_name" parent="parent_name"
reg=register_property_values
scsi-initiator-id=scsi_id;
```

For example, the file `/kernel/drv/isp.conf` for the system with SCSI ID 5 would resemble:

```
name="isp" parent="/sbus@1f,0"
reg=0x0,0x10000,1c2
scsi-initiator-id=5;
```

4. Shut down all systems in the cluster.

5. Cable the shared devices. Two-system clusters can be cabled with the devices between them, as illustrated in the figure below:



A Two-System Cluster Cabled with Devices Between Them

Disable SCSI termination on systems that are not positioned at the ends of the SCSI chain. Most single-ended SCSI controllers autodetect to disable termination, or are configured through software. Differential SCSI controllers typically require that you remove resistors from the controller card. For more information, refer to the documentation on controllers configured in your systems.

6. Boot each system:

   ```
   ok boot -r
   ```

   Watch for console messages from the driver changing the SCSI ID. For example:

   ```
   isp0: initiator SCSI ID now 5
   ```

7. Use the `prtvtoc`(1M) command to test the connection from each system to the shared devices.

   For example, type:

   ```
   # prtvtoc /dev/dsk/c1t0d0s0
   ```

   Output from this command confirms the connection between the system on which you typed the command and the disk.

Disabling the Abort Sequence on SPARC Systems

Sun SPARC systems provide console-abort sequences that enable you to halt and continue the processor:

- L1-A or STOP-A on the keyboard,

  or,

- BREAK on the serial console input device.

Each command is then followed by a response of "go" at the ok prompt to enable the system to continue.

VCS does not support continuing operations after the processor has been stopped by the abort sequence because data corruption may result. Specifically, when a system is halted with the abort sequence it stops producing heartbeats. The other systems in the cluster then consider the system failed and take over its services. If the system is later enabled with "go," it continues writing to shared storage as before, even though its applications have been restarted on other systems.

Disabling Keyboard Abort Sequence on Solaris 2.6

In Solaris 2.6, Sun introduced support for disabling the abort sequence. We recommend disabling the console-abort sequence on systems running Solaris 2.6 or greater. To do this:

1. Add the following line to the /etc/default/kbd file (create the file if it does not exist):

   ```
   KEYBOARD_ABORT=disable
   ```

2. Reboot.

3. If necessary, refer to the kbd(1) manual page for details.

Solaris 2.5.1

If a system has been stopped with the abort sequence in a VCS cluster, do not type go at the EEPROM prompt (ok). Instead, type boot at the ok prompt.

Installing VCS Using the InstallVCS Utility

`InstallVCS`, the VCS installation utility, is interactive. Using information you supply, it installs VCS on each system in the cluster, locally and remote, and sets up the VCS communication services, LLT and GAB. The local system need not be a part of the cluster.

`InstallVCS` does *not* install the VCS graphical user interface. See "Installing the VCS Cluster Manager" on page 30 for information about installing the `VRTScscm` package.

Note Do *not* use the `InstallVCS` utility if you are upgrading VCS. The `InstallVCS` utility is intended only for new installations of VCS. If you are upgrading, refer to the *VERITAS Cluster Server Release Notes 1.3.0* for procedures describing how to upgrade from your current level.

Preparation for Running Install Utility

Before you start installing VCS, be prepared to provide the following information:

- ✔ A name for the cluster; the name must begin with a letter of the alphabet (a-z, A-Z).
- ✔ A unique ID number for the cluster. Within the subnet containing the cluster, each cluster must have a unique ID. A new cluster using a duplicate cluster ID can cause existing clusters to fail.
- ✔ The host names of the systems in the cluster.
- ✔ The device names of the network interface cards used for the *private* networks among systems.

Also, make sure that when you run the installation utility:

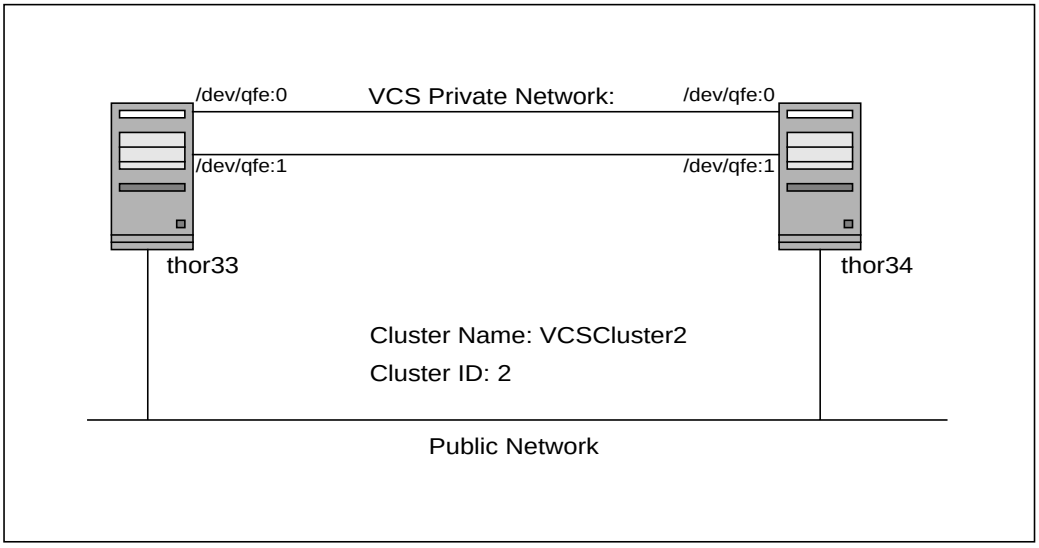
- ✔ You are logged in as superuser.
- ✔ You have remote root access to each of the systems in the cluster from the system where installation is run.
- ✔ Any packages or programs from previous installations of VCS — installed using `InstallVCS` or otherwise — have been *completely* removed from each system in the cluster. Refer to "Uninstalling VCS" on page 43.

Limitations of Private Network for InstallVCS

The `InstallVCS` utility sets up the LLT communications only on private networks that use direct physical network connections or use a hub. Do not use the `InstallVCS` utility where the cluster's private network employs switches.

Example System Installation

The following illustration shows two systems, thor33 and thor34, on which VCS is to run. For this example, the cluster's name is "VCSCluster2" and the Cluster's ID is "2". On each system, the private network uses the NIC devices `/dev/qfe:0` and `/dev/qfe:1`.



An Example Two-System Cluster on which VCS Will Be Installed

Using the VCS Interactive Installation Utility

The illustration above describes a cluster on which VCS is installed using the installation utility. The utility installs VCS over the network. You must have permissions to remotely log in as root on each of the systems to which VCS is to be installed.

Note If the user or the InstallVCS utility stops and exits the installation before completion, the partially installed VCS files must be removed before running the InstallVCS utility again. See "Uninstalling VCS" on page 43.

1. Log in as root user on system connected by the network to the systems where VCS is to be installed. The system from which VCS is installed does not need to be part of the cluster.

2. Insert the CD with the VCS software into a drive connected to the system.

   - If you are running Solaris volume-management software, the software automatically mounts the CD as `/cdrom/cdrom0`. Type the command:

     ```
     # cd /cdrom/cdrom0
     ```

   - If you are not running Solaris volume-management software, you must mount the CD manually. For example:

     ```
     # mount -F hsfs -o ro /dev/dsk/c0t6d0s2 /mnt
     ```

     Where, in this example, `/dev/dsk/c0t6d0s2` is the name for the CD drive.

     ```
     # cd /mnt
     ```

3. Start the VCS installation by entering:

   ```
   # ./InstallVCS
   ```

4. Enter information at the prompts, making sure the cluster name begins with a letter of the alphabet (a-z, A-Z) and that the cluster ID is unique number from 0 to 255:

   ```
   Please enter the unique Cluster Name : VCSCluster2
   Please enter the unique Cluster ID (a number from 0-255) : 2
   Enter the systems on which you want to install. (system names
         seperated by spaces) : thor33 thor34
   ```

5. The install process analyzes each system before proceeding.

```
Analyzing the system for install.
thor33 ..... OK
SunOS thor33 5.6 Generic_105181-20 sun4u sparc SUNW,Ultra-2
thor34 ..... OK
SunOS thor33 5.6 Generic_105181-20 sun4u sparc SUNW,Ultra-2
...
```

If the analysis reveals problems, such as insufficient disk space or the existence of previously installed VCS files, the installation stops. A message indicates the problem. See "Possible Problems with Installation" on page 20.

6. The process begins installing the VCS packages on each system. This could take a few minutes for each system:

```
Installing on thor33.

Copying VRTSperl binaries.
Installing  VRTSperl  Done.
Copying VRTSllt binaries.
Installing  VRTSllt  Done.
Copying VRTSgab binaries.
Installing  VRTSgab  Done.
Copying VRTSvcs binaries.
Installing  VRTSvcs  Done.

Installing on thor34.

Copying VRTSperl binaries.
Installing  VRTSperl  Done.
Copying VRTSllt binaries.
Installing  VRTSllt  Done.
Copying VRTSgab binaries.
Installing  VRTSgab  Done.
Copying VRTSvcs binaries.
Installing  VRTSvcs  Done.
```


7. The process now discovers information about the NICs (network interface cards) for setting up the VCS private network. You will see messages resembling the following:

```
Discovering NICs for setting up private network links on thor33.
This may take some time.

Following is the list of discovered NICs:
Sr. No.      NIC Device
1.           /dev/hme:0
2.           /dev/qfe:0
3.           /dev/qfe:1
4.           /dev/qfe:2
5.           /dev/qfe:3
6.           Other
```

   a. Select the device files that correspond to the *private* network links you have installed. (Note: Do not select the NICs used for the public network):

```
From the list above, please enter the serial number (the number
appearing in the "Sr. No." column) of the NIC for the
first PRIVATE network link: 2

From the list above, please enter the serial number (the number
appearing in the "Sr. No." column) of the NIC for the
second PRIVATE network link: 3
```

   b. If you must enter a NIC device other than those discovered, enter the number that corresponds to "Other." In this example, you enter **6**.

      Then, at the prompt, enter the complete pathname of the device.

8. Answer the question about the remaining systems:

```
Do you have the same network cards set up on all systems (Y/N)?
```

   If you answer yes ("Y"), the install process proceeds. See step 9 on page 19.

   If you answer no ("N"), for each of the cluster systems, one-by-one, the process prompts you to select the network cards from a list of NICs discovered on that system, just as in step 7.

9. As the installation proceeds, you will see messages confirming the installation. The following messages are typical.

```
Copied llt & gab configuration files.
Loading GAB and LLT modules and starting VCS on thor33:
Starting LLT
Start GAB
Loading GAB and LLT modules and starting VCS on thor34:
Starting LLT
Start GAB

Installation successful on all systems.
```

10. Systems running Solaris versions earlier than 2.6 will require rebooting. If you have such systems, you receive a message informing you that rebooting them is necessary.

```
Installation needs the following system/s to be rebooted in
order to start VCS.

Reboot system/s now (Y/N)?
```

   a. In most cases, you answer yes (“Y”). All applicable systems reboot, except the local system.

```
You are installing from a system in the cluster-
Skipping reboot on local system.
```

   After all other applicable systems in the cluster are rebooted, reboot the local system when you are prompted.

   b. If you answer no (“N”), you receive the message:

```
You chose not to reboot the system/s. Please note that you
need to reboot the above system/s in order to start VCS
properly.
```


Possible Problems with Installation

You may encounter problems installing VCS using the utility, `InstallVCS`. Refer to the following paragraphs for typical problems and their solutions.

Incorrect Permissions for Root on Remote System

The permissions are inappropriate. Make sure you have remote root access permission on each system to which you are installing.

```
....
Analyzing the system for install.
thor33 ..permission denied
RSH permissions not available on: thor33
```

Suggested solution: add a line in the file `/.rhosts` giving root access to the system running the install program. A "+" character in the first line of `/.rhosts` will permit any remote host to access the system.

Inaccessible System

The system you specified is not accessible. This could be for a variety of reasons, such as, the system name was entered incorrectly or the system is not available over the network.

```
...
Analyzing the system for install.
thor33 .The specified system does not exist or is inaccessible.
```

Suggested solution: verify that you entered the system name correctly; use the `ping`(1M) command to verify the accessibility of the host;.

VCS Files From a Previous Installation Exist

A VCS package from a previous installation already exists.

```
Analyzing the system for install.
thor33 ..... OK
SunOS thor33 5.6 Generic_105181-20 sun4u sparc SUNW,Ultra-2
thor34 ..... ERROR
VRTSllt package already installed.
...
```

Suggested solution: remove the package listed in the error message, and any other VCS files previously installed. Refer to "Uninstalling VCS" on page 43.

VCS, LLT, or GAB Process Running

If you use the `InstallVCS` utility to install VCS to a system where VCS or the utilities LLT or GAB are currently running, the installation stops with an error that resembles:

```
Analyzing the system for install.
thor34 ..... ERROR
GAB found running
```

This can occur if you try to reinstall VCS without completely removing VCS or its utilities from the previous installation.

Suggested solution: uninstall VCS completely. Refer to "Uninstalling VCS" on page 43.

Temp Directory Not Accessible on Local System

The `InstallVCS` utility requires access to `/tmp` on the local system. If the directory does not exist or is inaccessible because of the permissions, `InstallVCS` stops with an error that resembles:

```
Analyzing the system for install.
thor33 ..... ERROR
The /tmp directory is inaccessible on the local machine.
```

Suggested solution: check whether the `/tmp` directory exists; create `/tmp` if it does not. Make sure permissions allow installation to proceed.

Invalid Cluster Name Causes Stale Configuration

If you begin a cluster name with any other character (a number, for example) other than an alphabetical character, a-z or A-Z, the installation of VCS will appear to succeed, but all VCS configurations will report as "stale."

You can verify that this is the problem by typing the command from the directory `/etc/VRTSvcs/conf/config`:

```
# hacf -verify .
```

You receive messages that report that the cluster name is specified in error. For example:

```
VCS:12123: Type '_CLUSTER' defined more than once

VCS:12095: Aborting hacf: errors encountered
```

Suggested solution: you must change the cluster name to a valid name. This can be done by editing the newly created VCS configuration file and renaming the cluster:

1. Stop VCS on all machines:

   ```
   # hastop -all
   ```

2. Log on to one of the machines in the cluster.

3. Edit the VCS configuration file, `/etc/VRTSvcs/conf/config/main.cf`. (See "main.cf" on page 24). You can use `vi` or another text editor. Look for the line that contains the keyword "cluster" followed by the invalid name; correct the cluster name.

4. Save and close the file.

5. Enter the command:

   ```
   # /sbin/hastart
   ```

   VCS will restart on the system in a RUNNING state. See "Verifying the Cluster" on page 28 for more information.

6. Log on to another system in the cluster and perform step 5. VCS starts up, using the revised configuration file. Repeat this step on each of the remaining systems in the cluster.

Verifying the Installation

After successful installation, you can inspect the contents of the key configuration files that have been installed and modified during the process. These files reflect the configuration based on the information you supplied.

LLT and GAB Configuration Files

These files are required by the VCS communication services, LLT (Low Latency Transport) and GAB (Group Membership and Atomic Broadcast).

/etc/llthosts

The file `llthosts(4)` is a database, containing one entry per system, that links the LLT system ID (in the first column) with the LLT host name. This file is identical on each system in the cluster.

For example, the file `/etc/llthosts` contains entries that resemble:

```
0  thor33
1  thor34
```

/etc/lltab

The file `lltab`(1M) contains information that is derived during installation and used by the utility `lltconfig`(1M). After installation, this file lists the network links that correspond to the specific system.

For example, the file `/etc/lltab` contains entries that resemble:

```
set-node thor33
set-cluster 2
link link0 /dev/qfe:0 - ether - -
link link0 /dev/qfe:1 - ether - -
start
```

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, beginning with the `link` command, identify the two network cards that will be used by the LLT protocol. The last line starts the LLT protocol with the `start` command.

Refer to the `lltab`(4) manual page for details about how the LLT configuration may be modified. The directives in the `lltab` file must follow in an order as described in the manual page.

/etc/gabtab

After installation, the file `/etc/gabtab` contains a `gabconfig`(1M) command that configures the GAB driver for use.

The file `/etc/gabtab` should contain entries that resemble:

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

Where the `-c` option configures the driver for use and `-n2` specifies automatic seeding for a cluster of two systems.

main.cf

The VCS configuration file `/etc/VRTSvcs/conf/config/main.cf` is created during the installation process. A typical file after initial installation resembles:

```
include "types.cf"
cluster VCSCluster2 (
   UserNames = { admin = "cDRpdxPmHpzS." }
)
system thor33

system thor34
```

After installation, the `main.cf` file contains the minimum information that defines the cluster and its systems. In addition, the file `types.cf`, which is listed in the `include` statement, defines the VCS bundled agents for VCS resources. The file `types.cf` is also located in the directory `/etc/VRTSvcs/conf/config` after installation.

Notice that the cluster has an attribute UserNames. The `InstallVCS` utility creates a user, “admin,” whose password is encrypted; this password is the word: “password.”

Refer to the *VERITAS Cluster Server User’s Guide* and review the chapter on “The VCS Configuration Language.” The section “Sample Configuration Files” includes more extensive examples of a `main.cf` and `types.cf` for UNIX systems.

Verifying LLT, GAB, and Cluster Operation

Before attempting to verify LLT, GAB, or the cluster, you must:

- ✔ Log in to any system in the cluster as `root`.
- ✔ Place the VCS command directory in your PATH variable:

```
# export PATH=$PATH:/sbin:/usr/sbin:/opt/VRTSvcs/bin
```

Verifying LLT

Use the `lltstat` command to verify that links are active for LLT. This command returns information about the links for LLT for the system on which it is typed. Refer to the `lltstat`(1M) manual page for more information. In the following example, `lltstat -n` is typed on each system in the cluster.

System 1

```
# lltstat -n
```

Output resembles:

```
LLT node information:
    Node     State       Links
   *0        OPEN        2
    1        OPEN        2
sys1#
```

System 2

```
# lltstat -n
```

Output resembles:

```
LLT node information:
    Node     State       Links
    0        OPEN        2
   *1        OPEN        2
sys2#
```

Note that each system has two links and that each system is in the OPEN state. The asterisk (*) denotes the system on which the command is typed.

If the output of `lltstat -n` does not show all the systems in the cluster or two links for each system, use the verbose option of `lltstat`. For example, type `lltstat -nvv | more` on a system to view additional information about LLT.

In the following example, `lltstat -nvv | more` is typed on a system in the cluster.

System 1

```
# lltstat -nvv | more
```

Output resembles:

```
Node      State       Link     Status    Address
0         CONNWAIT
                      qfe0     DOWN
                      qfe1     DOWN
*1        OPEN
                      qfe0     UP        08:00:20:93:0E:34
                      qfe1     UP        08:00:20:93:0E:34
2         OPEN
                      qfe0     UP        08:00:20:8F:D1:F2
                      qfe1     UP        08:00:20:8F:D1:F2
3         CONNWAIT
                      qfe0     DOWN
                      qfe1     DOWN
```

Note that each system should be OPEN, each link should be UP, and each address should be correct. If the output of `lltstat` indicates otherwise, LLT is not operating. In this case, LLT is configured incorrectly.

To obtain information about the ports open for LLT, type `lltstat -p` on any system. In the following example, `lltstat -p` is typed on one system in the cluster.

System 1

```
# lltstat -p
```

Output resembles:

```
LLT port information:
    Port    Usage       Cookie
   0        gab         0x0
         opens:      0 1 3 4 5 6 7 8 9 10 11 12 13...
         connects:   0 1
sys1#
```

Note that two systems (0 and 1) are connected.

Verifying GAB

To verify that GAB is operating, type the following command on each system:

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

If GAB is operating, the following GAB port membership information is returned:

```
GAB Port Memberships
===================================
Port a gen a36e0003 membership 01
Port h gen fd570002 membership 01
```

`Port a` indicates that GAB is communicating, `gen a36e0003` is a random generation number, and `membership 01` indicates that systems `0` and `1` are connected.

`Port h` indicates that VCS is started, `gen fd570002` is a random generation number, and `membership 01` indicates that systems `0` and `1` are both running VCS.

If GAB is not operating, no GAB port membership information is returned:

```
GAB Port Memberships
===================================
```

If only one network is connected, the following GAB port membership information is returned:

```
GAB Port Memberships
===================================
Port a gen a36e0003 membership 01
Port a gen a36e0003 jeopardy    1
Port h gen fd570002 membership 01
Port h gen fd570002 jeopardy    1
```

For more information on GAB, refer to the *VERITAS Cluster Server User's Guide*.

Verifying the Cluster

To verify that the cluster is operating, type the following command:

```
# hastatus -summary
```

The output resembles:

```
-- SYSTEM STATE
-- System               State                Frozen

A  thor33               RUNNING              0
A  thor34               RUNNING              0
```

Note the system state. If the value is RUNNING, VCS is successfully installed and running. Refer to `hastatus`(1M) manual page. In the *VERITAS Cluster Server User's Guide*, Appendix B, "Advanced Topics," describes system states and the transitions between them.

Also, on one of the systems, type the `hasys`(1M) command:

```
# /opt/VRTSvcs/bin/hasys -display
```

The output resembles:

```
#System   Attribute          Value
thor33    AgentsStopped      0
thor33    ConfigBlockCount   54
thor33    ConfigCheckSum     29776
thor33    ConfigDiskState    CURRENT
thor33    ConfigFile         /etc/VRTSvcs/conf/config
thor33    ConfigInfoCnt      0
thor33    ConfigModDate      Wed Dec 31 23:00:00 1999
thor33    DiskHbDown
thor33    Frozen             0
thor33    GUIIPAddr
thor33    LLTNodeId          0
thor33    LinkHbDown
thor33    Load               0
thor33    LoadRaw            runque 0 memory 0 disk 0...
thor33    MajorVersion       1
thor33    MinorVersion       17
thor33    NodeId             0
thor33    OnGrpCnt           0
thor33    ShutdownTimeout    60
thor33    SourceFile         ./main.cf
thor33    SysName            thor33
thor33    SysState           RUNNING
thor33    TFrozen            0
thor33    TRSE               0
thor33    UpDownState        Up
thor33    UserInt            0
thor33    UserStr
```

On each system, the output should be similar. For more information on the `hasys -display` command, refer to the `hasys`(1M) manual page. Also refer to the chapter in the *VCS Cluster Server User's Guide*, "Administering VCS From the Command Line."

Installing the VCS Cluster Manager

You can use the VCS graphical user interface, Cluster Manager, to administer VCS. After VCS has been installed, you can install Cluster Manager on a UNIX system with X-Windows, or on a Windows NT workstation. The UNIX system from which you run Cluster Manager can be a system in the cluster or a remote workstation; the latter enables each system in the cluster to be administered remotely.

For information about using Cluster Manager, see the chapter "About the VCS GUI" in the *VERITAS Cluster Server User's Guide.*

Installing the Cluster Manager on UNIX (Solaris)

1. Insert the CD with the VCS software into a drive connected to the system.
   - If you are running Solaris volume-management software, the software automatically mounts the CD as `/cdrom/cdrom0`. Type the following command:

     # **`cd /cdrom/cdrom0`**
   - If you are not running Solaris volume-management software, you must mount the CD manually. For example:

     # **`mount -F hsfs -o ro /dev/dsk/c0t6d0s2 /mnt`**

     Where, in this example, `/dev/dsk/c0t6d0s2` is the name for the CD drive.

     # **`cd /mnt`**
2. Type the command:

   # **`pkgadd -d VRTScscm`**
3. Answer Yes if prompted.

Installing the Cluster Manager on a Windows NT Workstation

If you are installing the VCS Cluster Manager on a Windows NT workstation to administer the cluster, do the following:

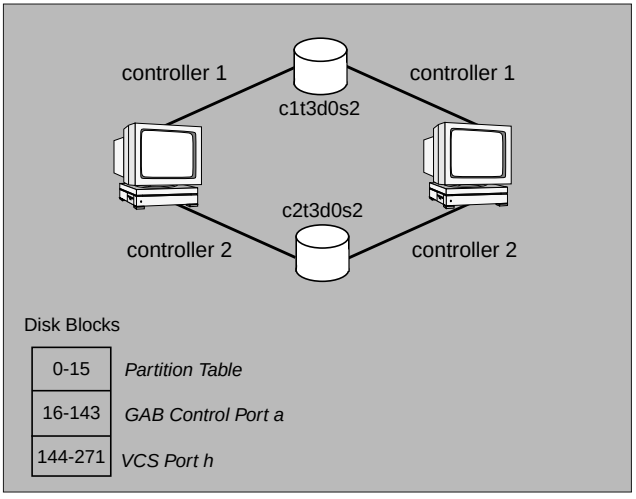
1. Insert the VCS CD into the drive on your Windows NT workstation.
2. Using Window Explorer, select the CD drive.
3. Go to \ntgui\Disk1.
4. Double-click Setup.exe.
5. The VCS InstallShield guides you through the installation process.

Configuring Membership Heartbeat Regions on Disk

You can set up membership heartbeat regions on shared disks for use as an additional path for VCS heartbeating (see“Two Types of Channels: Network and Shared Disks” on page 5. With these regions configured in addition to network connections, VCS has multiple heartbeat paths available. If one network connection fails, VCS has the remaining network connections and the heartbeat disk region that allow heartbeating to continue.

When you configure the membership heartbeat regions on a disk, you must set up two regions, each consisting of 128 blocks: one for the GAB control (port a) and one for the VCS (port h).

In the following illustration, two systems are connected by two shared disks. Each system uses a separate controller for each disk. For example, partition 2 of disk target number 3 could be allocated for heartbeating on both shared I/O chains. Because partition 2 begins on the first cylinder of the disk, the regions must start on or after block 16 because the partition table uses blocks 0 through 15.



Allocation of Heartbeat Disk Regions

Editing the /etc/gabtab File to Add Heartbeat Regions

You can set up heartbeat regions within a disk partition by adding `gabdiskhb`(1M) commands to the `/etc/gabtab` file. The entries specify the block device name for the disk partition, the starting block number of the heartbeat region, and the port designation; a region (or several) may be associated with the GAB control port (port a) and the VCS communications port (port h).

The illustrated configuration is specified in a `/etc/gabtab` file that resembles:

```
gabdiskhb -a /dev/dsk/c1t3d0s2 -s 16 -p a
gabdiskhb -a /dev/dsk/c1t3d0s2 -s 144 -p h
gabdiskhb -a /dev/dsk/c2t3d0s2 -s 16 -p a
gabdiskhb -a /dev/dsk/c2t3d0s2 -s 144 -p h
gabconfig -c -n 2
```

The `-s` option to the `gabdiskhb` command specifies the start location of each 128-block region.

The `-p` option specifies the port: the value “a” specifies the GAB control port, and the value “h” specifies the VCS port.

The regions should not overlap. Two adjacent regions must have starting blocks separated by 128 blocks.

Usually, the first 16 blocks of the first partition of the disk are reserved. If the partition is not the first partition on the disk, the start locations may be 0 and 128.

Note the following considerations when configuring heartbeat disk regions.

- A disk partition containing a heartbeat region cannot be used for any other purpose, such as a file system or volume.
- If a disk containing heartbeat regions is also used by for other purposes, the traffic could adversely affect performance of the heartbeating.

Adding GAB Disk Region Signatures (Optional) for Integrity

To guarantee the integrity of the GAB disk region, GAB can be directed to verify a signature in that region on a periodic basis. This optional feature ensures that valuable data on the disk, such as a filesystem, is not accidently overwritten.

You can use the `gabdiskconf`(1M) command to initialize the region with the specified signature. This must be done before the `gabdiskhb` command is run manually or from the `/etc/gabtab` file during boot.

Example, Configuring and Checking for a Signature

In the following example, GAB disk regions are initialized by assigning signatures.

```
gabdiskconf -i /dev/dsk/c1t1d2s3 -s 16 -S 1123
gabdiskconf -i /dev/dsk/c1t1d2s3 -s 144 -S 1124
```

The disk regions, starting at block 16 and 144 of the block device `/dev/dsk/c1t1d2s3`, are assigned the 4-byte strings of 1123 and 1124, respectively, as signatures.

Later, the regions are configured as heartbeating regions by the `gabdiskhb` command. In the following example, the `gabdiskhb` command specifies that GAB check the signatures on a periodic basis.

```
gabdiskhb -a /dev/dsk/c1t1d2s3 -s 16 -p a -S 1123
gabdiskhb -a /dev/dsk/c1t1d2s3 -s 144 -p h -S 1124
```

If GAB determines that a signature does not match the user's specified value, it marks the disk as faulted.

Initializing File Systems and Disk Groups on Shared Storage

In addition to the shared disk partitions used for VCS communications, your configuration may include disks on the shared bus that contain VERITAS Volume Manager™ (VxVM) disk groups or file systems. You must initialize these disk groups and file systems from one system only.

For VxVM configurations, install VxVM as instructed in the *VERITAS Volume Manager Installation Guide*. Disks on the shared bus must be configured into disk groups other than `rootdg`. Create disk groups on one system only. They will be deported and imported onto the other system by VCS as necessary. Similarly, use `mkfs` to make shared file systems from one system only. They will be mounted on other systems by VCS as necessary.

Note Do not add exported file systems to `/etc/vfstab` or `/etc/dfs/dfstab`. VCS will mount and export these file systems automatically.

Configuring Disk Regions on Volume Manager Disks

Communication disk regions and service group heartbeat disk regions can coexist on a disk controlled by VxVM. However, these disk regions cannot be configured on VxVM volumes, and must be configured instead on the block ranges of the underlying physical device. The space for these partitions must be allocated before a disk is initialized by the VxVM.

Follow the steps below to prepare a disk for VCS communication and VxVM storage:

1. Install VxVM as instructed in the *VERITAS Volume Manager Installation and Configuration Guide*.
2. Identify the disk by its VxVM tag name, for example, `c1t1d0`.
3. If the disk contains data, migrate the data to another storage media.
   a. Unmount all file systems on the disk.
   b. Remove any volumes, plexes, or subdisks from the disk.
   c. Remove the disk from any active disk group or deport its disk group.

4. On any system, place the VCS command directory in your path.
For example:

```
# export PATH=$PATH:/opt/VRTSvcs/bin
```

5. Allocate a VCS partition on the disk. Type:

```
# hahbsetup disk_tag
```

Enter `y` when prompted. The `hahbsetup` command sets up disk communication for VxVM and VCS. The variable *disk_tag* refers to the name you identified in step 2. For example:

```
# hahbsetup c1t1d0
```

Output resembles:

```
The hadiskhb command is used to set up a disk for combined use
by VERITAS Volume Manager and VERITAS Cluster Server for disk
communication.
WARNING: This utility will destroy all data on c1t1d0
Have all disk groups and file systems on disk c1t1d0 been either
unmounted or deported? y
There are currently slices in use on disk /dev/dsk/c1t1d0s2
Destroy existing data and reinitialize disk? y
1520 blocks are available for VxCS disk communication and
service group heartbeat regions on device /dev/dsk/c1t1d0s7
This disk can now be configured into a Volume Manager disk
group. Using vxdiskadm, allow it to be configured into the disk
group as a replacement disk. Do not select reinitialization of
the disk.
After running vxdiskadm, consult the output of prtvtoc to
confirm the existence of slice 7. Reinitializing the disk
under VxVM will delete slice 7. If this happens, deport the disk
group and rerun hahbsetup.
```

6. The disk should now be initialized, even though it has not been added to a disk group. To add the disk to a disk group, run the `vxdg addisk` command (refer to the `vxdg` manual page for more information). For example, after running `hahbsetup` to allocate a VCS partition on `clt1d0`, add `clt1d0` to the `sharedg` disk group as `disk01` by typing the following command:

   # **`vxdg -g sharedg adddisk disk01=clt1d0`**

7. Display the partition table. Type:

   # **`prtvtoc /dev/dsk/`***`disk_tags0`*

   For example:

   # **`prtvtoc /dev/dsk/c1t1d0s0`**

   Output resembles:

```
Partition   Tag    Flags   First      Sector      Last       Mount
                           Sector     Count       Sector     Directory
2           5      01      0          8887440     8887439
3           15     01      0          1520        1519
4           14     01      3040       8884400     8887439
7           13     01      1520       1520        3039
```

8. Confirm that slice 7 exists and that its tag is 13.

9. Configure partition `/dev/dsk/c1t1d0s7` into VCS.

Preparing NFS Services

Block Devices

Your configuration may include disks on the shared bus that support NFS. File systems exported by NFS can be configured on disk partitions or on VERITAS Volume Manager volumes. An example disk partition name is `/dev/dsk/c1t1d0s3`. An example volume name is `/dev/vx/dsk/shareydg/vol3`. Each name represents the block device on which the file system will be mounted.

Major and Minor Numbers

Block devices providing NFS service must have the same major and minor numbers on each system. Major and minor numbers are used by Solaris to identify the logical partition or disk slice. NFS also uses them to identify the exported file system. Major and minor numbers must be checked to ensure that the NFS identity for the file system is the same when exported from each system.

Checking Major and Minor Numbers

1. Use the following command on all systems exporting an NFS file system. This command displays the major and minor numbers for the block device. For VxVM volumes, you must first import the associated shared disk group on each system.

   ```
   # ls -lL block_device
   ```

   The variable *block_device* refers to a partition on which a file system is mounted for export via NFS. Use this command on each NFS file system. For example, type:

   ```
   # ls -lL /dev/dsk/c1t1d0s3
   ```

   Output on System A resembles:

   ```
   crw-r-----  1 root  sys  32,134 Dec 3 11:50 /dev/dsk/c1t1d0s3
   ```

   Output on System B resembles:

   ```
   crw-r-----  1 root  sys  32,134 Dec 3 11:55 /dev/dsk/c1t1d0s3
   ```

   Note that the major numbers, 32, and the minor numbers, 134, match.

2. If either the major or the minor numbers do not match, refer to "If Major and Minor Numbers Do Not Match" in the next section.

3. Check major and minor numbers on each block device used for NFS.

If Major and Minor Numbers Do Not Match

1. Place the VCS command directory in your path. For example:

   ```
   # export PATH=$PATH:/opt/VRTSvcs/bin
   ```

2. If the block device is a volume, identify on each system the two major numbers used by the VERITAS Volume Manager:

   ```
   # grep vx /etc/name_to_major
   ```

   Output on System A would resemble:

   ```
   vxio 32
   vxspec 33
   ```

   On System B:

   ```
   vxio 36
   vxspec 37
   ```

3. Type the following command on System B to change the major number (36/37) to match that of System A (32/33):

   For disk partitions:

   ```
   # haremajor -sd major_number
   ```

   For volumes:

   ```
   # haremajor -vx major_number1 major_number2
   ```

   The variable *major_number* represents the numbers from System A.

   For example, for disk partitions:

   ```
   # haremajor -sd 32
   ```

   For volumes:

   ```
   # haremajor -vx 32 33
   ```

   If this command fails, you will receive a report similar to the following:

   ```
   Error: Preexisiting major number 32
   These are available numbers on this system: 128...
   Check /etc/name_to_major on all systems for
   available numbers.
   ```


4. If you receive this report, type the following command on System A to change the major number (32/33) to match that of System B (36/37):

   For disk partitions:

   ```
   # haremajor -sd 36
   ```

   For volumes:

   ```
   # haremajor -vx 36 37
   ```

   If the command fails again, you will receive a report similar to the following:

   ```
   Error: Preexisiting major number 36
   These are available numbers on this node: 126...
   Check /etc/name_to_major on all systems for
   available numbers.
   ```

5. If you receive the second report, choose the larger of the two available numbers (in this example, 128), and use this number in the `haremajor` command to reconcile the major numbers. Type the following command on both systems:

   For disk partitions:

   ```
   # haremajor -sd 128
   ```

   For volumes:

   ```
   # haremajor -vx 128 129
   ```

6. Reboot each system on which `haremajor` was successful.

7. If the minor numbers match, proceed to reconcile the major and minor numbers of your next partition.

8. If the block device on which the minor number does not match is a volume, consult the `vxdg`(1M) manual page for instructions on reconciling the VERITAS Volume Manager minor numbers, with specific reference to the `reminor` option.

9. For disk partitions, complete steps a–e, below. (In this example, the minor numbers are 134 and 62.)

   a. Type the following command on both systems using the name of your block device:

      ```
      # ls -1 /dev/dsk/c1t1d0s3
      ```

      Output from this command resembles the following on System A:

      ```
      lrwxrwxrwx  1 root  root  83 Dec 3 11:50
               /dev/dsk/c1t1d0s3 -> ../../
               devices/sbus@1f,0/QLGC,isp@0,1000/sd@2,0:d,raw
      ```

      The device name (in bold, above) includes the slash following the word devices, and continues to, but does not include, the colon.

   b. Type the following command on both systems to determine the instance numbers used by the SCSI driver:

      ```
      # grep sd /etc/path_to_inst | sort -n -k 2,2
      ```

      Output from this command resembles the following on System A:

      ```
      "/sbus@1f,0/QLGC,isp@0,10000/sd@0,0" 0 "sd"
      "/sbus@1f,0/QLGC,isp@0,10000/sd@1,0" 1 "sd"
      "/sbus@1f,0/QLGC,isp@0,10000/sd@2,0" 2 "sd"
      "/sbus@1f,0/QLGC,isp@0,10000/sd@3,0" 3 "sd"
      "/sbus@1f,0/QLGC,isp@0,10000/sd@4,0" 4 "sd"
      "/sbus@1f,0/QLGC,isp@0,10000/sd@5,0" 5 "sd"
      "/sbus@1f,0/QLGC,isp@0,10000/sd@6,0" 6 "sd"
      "/sbus@1f,0/QLGC,isp@0,10000/sd@8,0" 7 "sd"
      "/sbus@1f,0/QLGC,isp@0,10000/sd@9,0" 8 "sd"
      "/sbus@1f,0/QLGC,isp@0,10000/sd@a,0" 9 "sd"
      "/sbus@1f,0/QLGC,isp@0,10000/sd@b,0" 10 "sd"
      "/sbus@1f,0/QLGC,isp@0,10000/sd@c,0" 11 "sd"
      "/sbus@1f,0/QLGC,isp@0,10000/sd@d,0" 12 "sd"
      "/sbus@1f,0/QLGC,isp@0,10000/sd@e,0" 13 "sd"
      "/sbus@1f,0/QLGC,isp@0,10000/sd@f,0" 14 "sd"
      "/sbus@1f,0/SUNW,fas@e,8800000/sd@0,0" 15 "sd"
      "/sbus@1f,0/SUNW,fas@e,8800000/sd@1,0" 16 "sd"
      "/sbus@1f,0/SUNW,fas@e,8800000/sd@2,0" 17 "sd"
      "/sbus@1f,0/SUNW,fas@e,8800000/sd@3,0" 18 "sd"
      "/sbus@1f,0/SUNW,fas@e,8800000/sd@4,0" 19 "sd"
      "/sbus@1f,0/SUNW,fas@e,8800000/sd@5,0" 20 "sd"
      "/sbus@1f,0/SUNW,fas@e,8800000/sd@6,0" 21 "sd"
      "/sbus@1f,0/SUNW,fas@e,8800000/sd@8,0" 22 "sd"
      "/sbus@1f,0/SUNW,fas@e,8800000/sd@9,0" 23 "sd"
      ```


```
"/sbus@1f,0/SUNW,fas@e,8800000/sd@a,0" 24 "sd"
"/sbus@1f,0/SUNW,fas@e,8800000/sd@b,0" 25 "sd"
"/sbus@1f,0/SUNW,fas@e,8800000/sd@c,0" 26 "sd"
"/sbus@1f,0/SUNW,fas@e,8800000/sd@d,0" 27 "sd"
"/sbus@1f,0/SUNW,fas@e,8800000/sd@e,0" 28 "sd"
"/sbus@1f,0/SUNW,fas@e,8800000/sd@f,0" 29 "sd"
```

c. Locate the device names in the output of step b, and identify the instance numbers that appear as the second field in each line. In this example, the device name on System A is `/sbus@1f,0/QLGC,isp@0,10000/sd@1,0`. The associated instance number is `1`.

d. Compare instance numbers.

- If the instance number from one system is not used on the other (that is, it does not appear in the output of step b), edit `/etc/path_to_inst` to make the second system's instance number equal to that of the first system.
- If the instance numbers are being used on both systems, edit `/etc/path_to_inst` on both systems. Change the instance number associated with the device name to an unused number greater than the highest number used by other devices. The output of step b shows the instance numbers used by all devices.

e. Type the following command to reboot each system on which `/etc/path_to_inst` was modified:

```
# reboot -- -rv
```

Uninstalling VCS

You can uninstall VCS using `pkgrm` command, or the `UnInstallVCS` utility, which is available on the CD-ROM on which VCS is supplied. Before removing VCS from any system in the cluster, shutdown and remove applications such as the Cluster Manager or any VCS enterprise agents that depend on VCS.

Note Do not attempt to uninstall VCS if you are running another application that uses the VCS modules GAB or LLT. Refer to the documentation supplied with that application for instructions on how to shut down the application before uninstalling VCS or any of its utilities.

Uninstalling VCS Using pkgrm

1. Shut down VCS on the local system using the `hastop`(1m) command.

   ```
   # hastop -local
   ```

2. Stop the GAB and LLT utilities.

   ```
   # gabconfig -U
   # lltconfig -U
   ```

3. Unload the GAB and LLT kernel modules.

   a. Use them `modinfo`(1m) and `grep`(1M) to determine the module IDs. The `modinfo` command indicates the ID of the module in the left-hand column

      ```
      # modinfo | grep GAP
      # modinfo | grep LLT
      ```

   b. Unload the modules from the kernel, using `modunload`(1M).

      ```
      # modunload -i ID_GAB
      # modunload -i ID_LLT
      ```

4. In the directory `/etc/VRTSvcs`, use `pkgrm`(1M) to remove the VCS packages in the following order:

   ```
   # pkgrm VRTSperl
   # pkgrm VRTSvcs
   # pkgrm VRTSgab
   # pkgrm VRTSllt
   ```

5. Repeat step 1 and step 4 for each system from which to uninstall VCS.

Uninstalling VCS Using the UnInstallVCS Utility

Before removing VCS from any system in the cluster, shutdown and remove applications such as the Cluster Manager or any VCS enterprise agents that depend on VCS.

Note If you are uninstalling VCS after an incomplete installation, you can expect to receive warnings about the inconsistent state of VCS. You can proceed with uninstalling VCS.

Caution Before running the `UnInstallVCS` utility, issue the command to gracefully stop VCS:

```
# hastop -all
```

1. Insert the CD with the VCS software into a drive connected to the system.
   - If you are running Solaris volume-management software, the software automatically mounts the CD as `/cdrom/cdrom0`. Type the command:

     ```
     # cd /cdrom/cdrom0
     ```

   - If you are not running Solaris volume-management software, you must mount the CD manually. For example:

     ```
     # mount -F hsfs -o ro /dev/dsk/c0t6d0s2 /mnt
     ```

     Where, in this example, `/dev/dsk/c0t6d0s2` is the name for the CD drive.

     ```
     # cd /mnt
     ```

2. Start the `UnInstallVCS` utility:

   ```
   # ./UnInstallVCS
   ```

3. Answer the prompt with the names of the systems on which to uninstall VCS:

   ```
   Enter the systems on which you want to uninstall. (system names
         seperated by spaces) : thor33 thor34

   Analysing the system for uninstall.
   thor33 ..... OK.

   thor34 ..... OK.
   ```

4. The VCS packages found on the systems are listed and removed.

Installing VCS Using pkgadd 3

You can use the `pkgadd` command to install the VCS packages on each of the systems instead of the installation utility, `InstallVCS.`

Using the `pkgadd` utility is appropriate when:

- You are installing a single VCS package.
- You are installing VCS to a single system in a cluster running VCS 1.2.1.
- You are unable to install to a system over the network. This can occur if the user does not have remote root user access.

▼ To install VCS using pkgadd:

- ✔ On each system, run pkgadd from the CD containing VCS software (see "Installing VCS On Each System Using pkgadd").
- ✔ On each system, configure LLT. See "Configuring Low Latency Transport (LLT)".
- ✔ On each system, configure GAB. See "Configuring Group Membership and Atomic Broadcast (GAB)".
- ✔ On each system, create a `main.cf` file.
- ✔ Reboot each system.
- ✔ Verify LLT, GAB, and Cluster operation. See "Verifying LLT, GAB, and Cluster Operation" on page 25.

Installing VCS On Each System Using pkgadd

On each system, perform the following steps to install the VCS software.

1. Insert the CD into a drive connected to your system.
   - If you are running Solaris volume-management software, the software automatically mounts the CD as `/cdrom/cdrom0`. Type the command:

     ```
     # cd /cdrom/cdrom0
     ```

   - If you are not running Solaris volume-management software, you must mount the CD manually. For example:

     ```
     # mount -F hsfs -o ro /dev/dsk/c0t6d0s2 /mnt
     ```

   Where, in this example, `/dev/dsk/c0t6d0s2` is the default name for the CD drive.

   ```
   # cd /mnt
   ```

2. Add the VCS packages.

   a. On each system, type the following command:

   ```
   # pkgadd -d .
   ```

   You will receive a message listing the available packages.

   b. When prompted, select the following in the order shown below:

   ```
   VRTSllt
   VRTSgab
   VRTSvcs
   VRTSperl
   ```

3. As the packages are being added, answer `Yes` when prompted.

Note If you want to add Cluster Manager (the VCS graphical user interface package), VRTScscm, you can do it after VCS is installed. See "Installing the VCS Cluster Manager" on page 30.

4. After all the packages are added, type `q`.

Caution Do not reboot at this time. Before rebooting, you must configure the VCS services, LLT and GAB. Proceed to the next section, "Configuring LLT and GAB."

Configuring LLT and GAB

LLT and GAB are used by VCS. They replace the functions of TCP/IP for VCS private network communications. LLT and GAB provide the performance and reliability required by VCS for these and other functions.

LLT and GAB must be configured as described in the following sections.

Configuring Low Latency Transport (LLT)

To configure LLT, set up two files: `/etc/llthosts` and `/etc/lltab` on each system in the cluster.

Setting Up /etc/llthosts

The file `llthosts(4)` is a database, containing one entry per system that links the LLT system ID (in the first column) with the LLT host name. You must create an identical file on each system in the cluster.

Using `vi`, or another editor, create the file `/etc/llthosts` that contains entries resembling:

```
1   thor33
2   thor34
```

Setting Up /etc/lltab

Each `/etc/lltab` file must specify the system's ID number, the network links that correspond to the system, and a command to start the LLT protocol. In addition, the file can contain other directives. See "LLT Directives" on page 48. Refer also to the sample `lltab` file in `/opt/VRTSllt`.

Using `vi`, or another editor, create the file `/etc/lltab` that contains entries that resemble:

```
set-node thor33
link qfe0 /dev/qfe:0 - ether - -
link qfe1 /dev/qfe:1 - ether - -
start
```

The first line identifies the system. The next two lines, beginning with the `link` command, identify the two private network cards that will be used by the LLT protocol. The last line starts the LLT protocol with the `start` command. The order of directives must be the same as in the sample file `/opt/VRTSllt/lltab`.

LLT Directives

`set-node`	Assigns the system ID. This number must be unique for each system in the cluster, and must be in the range 0-31. *Note that LLT fails to operate if any systems share the same ID.*
`link`	Attaches LLT to a network interface. At least one link is required, and up to eight are supported. 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` is the device name of the network interface. Its format is *device_name:device_instance_number*. 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.
`start`	The `start` directive must always appear last.
`set-cluster`	Assigns a unique cluster number. Use this directive when more than one cluster is configured on the same physical network connection. Note that LLT uses a default cluster number of zero.
`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. Note that LLT distributes network traffic evenly across all available network connections and, in addition to enabling VCS communication, broadcasts heartbeats to monitor each network connection.

For more information about LLT directives, refer to the `llttab`(4) manual page.

Additional Considerations for LLT

- Each network interface configured for LLT must be attached to a separate and distinct physical network.
- By default, Sun systems assign the same MAC address to all interfaces. Thus, connecting two or more interfaces to a network switch can cause problems. For example, if IP is configured on one public interface and LLT on another, and both interfaces are connected to a switch, the duplicate MAC address on the two switch ports can cause the switch to incorrectly redirect IP traffic to the LLT interface and vice-versa. To avoid this, configure the system to assign unique MAC addresses by setting the `eeprom`(1M) parameter `local-mac-address` to `true`.

Configuring Group Membership and Atomic Broadcast (GAB)

To configure GAB, use `vi` or another editor to set up an `/etc/gabtab` configuration file on each system in the cluster. Each `/etc/gabtab` file must specify the number of systems in the cluster. The following example shows a simple `gabtab`:

```
gabconfig -c -n 2
```

This configuration has no disks with heartbeat regions and expects two systems to join before VCS is seeded automatically.

To bypass protection from preexisting network partitions, replace the `-n` option with the `-x` option in the `gabconfig` command in `/etc/gabtab` on all systems. The resulting `gabtab` file is:

```
gabconfig -c -x
```


Creating a main.cf File

VCS requires the configuration file, `main.cf`, to exist in the VCS configuration directory `/etc/VRTSvcs/conf/config`. The `main.cf` configuration file has the following essential elements:

- An "include' statement that specifies the file, `types.cf`, which defines the VCS bundled agent resources.
- The name of the cluster.
- The name of the systems that make up the cluster.

Example, main.cf

```
include "types.cf"
cluster VCSCluster2 ( )
system thor33

system thor34
```

Note that the "include" statement refers to `types.cf`. This file, in which the VCS bundled agent resources are described, must be moved from the directory `/etc/VRTSvcs/conf/` into `/etc/VRTSvcs/conf/config`, the directory containing `main.cf`.

Configuring VCS

After the successful installation of VCS, you can modify the configuration of VCS by using the command line or by using Cluster Explorer (the VCS GUI).

Starting VCS

To start VCS, reboot each system in the cluster. During the reboot process, console messages about LLT and GAB are displayed. If LLT and GAB are configured correctly on each system, the output resembles:

```
.
.
.
LLT: link 0 node 1 active
LLT: link 1 node 1 active
.
.
.
VCS: starting on: thor3
VCS: waiting for configuration status
VCS: local configuration missing
VCS: registering for cluster membership
VCS: waiting for cluster membership
GAB: Port h gen 19fb0003 membership 01
VCS: received new cluster membership
VCS: all systems have stale configurations
.
.
.
```

Note that the links are `active` for LLT and that `Port h` is registered for GAB; therefore, LLT and GAB are configured correctly. Now, you must verify that the cluster is operating. Refer to “Verifying LLT, GAB, and Cluster Operation” on page 25.

If your network is not performing correctly, the output may resemble:

```
.
.
.
VCS: starting on: thor33
VCS: waiting for configuration status
VCS: local configuration missing
VCS: registering for cluster membership
GAB: Port h registration waiting for seed port membership
VCS: registration failed. Exiting
.
.
.
```

Note that `Port h` registration is incomplete. The network devices or cabling are configured incorrectly. Now, you must examine LLT, GAB, and the cluster for potential problems. Refer to “Verifying LLT, GAB, and Cluster Operation” on page 25

Index