

# VERITAS Cluster Server™ 2.0

---

## User's Guide

AIX

April 2002

30-000601-011

  
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

Copyright © 1998-2002 VERITAS Software Corporation. All Rights Reserved. VERITAS, VERITAS SOFTWARE, the VERITAS logo, VERITAS The Data Availability Company, and VERITAS Cluster Server are trademarks or registered trademarks of VERITAS Software Corporation in the USA and/or other countries. Other product names mentioned herein may be trademarks or registered trademarks of their

VERITAS Software Corporation  
350 Ellis Street  
Mountain View, CA 94043  
U.S.A.  
Phone 650-527-8000  
Fax 650-527-2908  
[www.veritas.com](http://www.veritas.com)



# Contents

---

<b>Preface</b>	<b>v</b>
How This Guide Is Organized	v
Technical Support	xi
Conventions	xi

## **Section I Overview of Cluster Technology**

---

<b>1. Getting Acquainted with Clustering</b>	<b>1</b>
What Clustering Offers	1
Can My Application be “Clustered”?	2
Detecting Failure	5
Failover and Switchover	6
Failover Configurations	7
Advanced Failover Configurations	10
<b>2. What is VERITAS Cluster Server?</b>	<b>11</b>
The Building Blocks of VCS	11
Fitting the Blocks Together	22
VCS Processes	24



---

## Section II VCS Configuration—The Basics

---

<b>3. Configuration Concepts</b>	<b>29</b>
The VCS Configuration Language	30
The main.cf File	30
The types.cf File	35
Attributes	37
Keywords/Reserved Words	41
Managing the VCS Configuration File	41
<b>4. The Role of Service Group Dependencies</b>	<b>43</b>
What is a Service Group Dependency?	43
Why Configure a Service Group Dependency?	44
Categories of Service Group Dependencies	45
Location of Dependency	47
Type of Dependency	49
Overview of Service Group Dependency Configurations	52
Configuring Service Group Dependencies	60
Automatic Actions for Service Group Dependencies	61
Manual Operations for Service Group Dependencies	63
Linking Service Groups (Online/Offline Dependencies)	64
Dependency Summary Sheet and FAQs	66

## Section III Administration—Putting VCS to Work

---

<b>5. Controlling Access to VCS</b>	<b>73</b>
User Privileges	73
Administration Matrices	77
User Privileges for CLI and Cluster Shell Commands	82



---

<b>6. Administering VCS from the Command Line</b>	<b>83</b>
VCS Environment Variables	83
How VCS Identifies the Local System	84
Installing a VCS License	85
Starting VCS	85
Stopping VCS	87
Adding, Modifying, and Deleting Users	89
Querying VCS	92
Basic Administrative Operations	99
Basic Configuration Operations	103
<b>7. Administering VCS from Cluster Manager (Java Console)</b>	<b>117</b>
Getting Started	117
Cluster Monitor	119
Cluster Explorer	126
Configuration Editor	136
Accessing Additional Features of the Java Console	137
Using Cluster Monitor	146
Logging On to and Off of a Cluster	148
Administering User Profiles	150
Administering Service Groups	154
Administering Resources	174
Importing Resource Types	188
Administering Systems	189
Administering the Cluster Configuration	193
Using Configuration Editor	196
Running Commands	198
Editing Attributes	199
Customizing the Log Display	201
Uninstalling Cluster Manager (Java Console)	202



---

<b>8. Administering VCS from Cluster Manager (Web Console)</b>	<b>203</b>
About the Web Console	203
Installation and Configuration	215
Before Using the Web Console	217
Connecting to the Web Console	217
Navigating the Web Console	218
Refreshing the Web Console	219
Administering Clusters	220
Administering Service Groups	221
Administering Resources	226
Administering Systems	228
Editing Attributes	230
Customizing the Log Display	233

## Section IV Advanced Topics—What VCS Can Do

---

<b>9. Service Group Workload Management</b>	<b>237</b>
Managing Load from Service Group and System Levels	237
Deciding Startup and Failover Locations	237
Additional Considerations	241
Sample Definitions and Configurations	243
<b>10. Notification</b>	<b>259</b>
How It Works	259
Notification Components	262
List of Events and Traps	264
Configuring Notification	269
<b>11. Event Triggers</b>	<b>271</b>
Event Trigger Notification	271
Event Triggers	272



---

<b>12. VCS Performance Considerations</b>	<b>281</b>
Impact of VCS on Overall System Performance	281
VCS Performance	285
Scheduling Class and Priority Configuration Support	291
<b>13. Troubleshooting and Recovery</b>	<b>293</b>
VCS Logging	293
Network Partitions and Split-Brain	295
Disabling Resources	311
VCS Utilities	314
Troubleshooting VCS Startup	316
Troubleshooting Service Groups	317
Troubleshooting Resources	320
Troubleshooting Notification	321
Troubleshooting Cluster Manager (Web Console)	322

## Section V Appendixes

---

<b>A. System States</b>	<b>329</b>
<b>B. VCS Attributes</b>	<b>333</b>
Resource Attributes	334
Resource Type Attributes	339
Service Group Attributes	344
System Attributes	353
Cluster Attributes	360
<b>C. Additional Administration Matrices</b>	<b>365</b>
haagent	365
haattr	366
hacli	366
haconf	367
hadebug	367



---

halog	368
hareg	368
hatype	369
<b>Index</b>	<b>371</b>



# Preface

---

## How This Guide Is Organized

Chapter 1, “[Getting Acquainted with Clustering](#),” presents the basics on clustering software, including failover detection and storage considerations. This chapter also includes information regarding first- and second-generation clustering packages, and tips on how to manage a cluster.

Chapter 2, “[What is VERITAS Cluster Server?](#),” explains the “building blocks” of VCS and how they interact with one another in a cluster environment, and introduces the core VCS processes.

Chapter 3, “[Configuration Concepts](#),” describes the VCS configuration language, including attributes, definitions, clauses, and dependencies. This chapter also includes a list of key and reserved words, and an overview of basic configuration concepts, such as the contents of the main.cf and types.cf configuration files.

Chapter 4, “[The Role of Service Group Dependencies](#),” defines the concept of a service group dependency and explains their default behavior. This chapter also explains the various categories, locations, and types of dependencies.

Chapter 5, “[Controlling Access to VCS](#),” describes the enhanced user-privilege model and provides matrices to determine which command options can be executed within a specific user category.

Chapter 6, “[Administering VCS from the Command Line](#),” provides instructions on how to perform basic and advanced administrative tasks from the command line.

Chapter 7, “[Administering VCS from Cluster Manager \(Java Console\)](#),” provides an overview of the VCS Java graphical user interface and configuration tool. This chapter also includes instructions on how to perform basic and advanced administrative tasks.

Chapter 8, “[Administering VCS from Cluster Manager \(Web Console\)](#),” provides an overview of the VCS Web-based graphical user interface and includes instructions on how to perform basic monitoring and administrative tasks.

Chapter 9, “[Service Group Workload Management](#),” explains the latest load balancing mechanism and how VCS employs this functionality at service group level. This chapter also describes the concept of “static load”; specifically, how to statically associate system capacity and group load within the main configuration file, main.cf.



Chapter 10, “[Notification](#),” explains how VCS uses SNMP and SMTP to notify administrators of important events, such as a resource or system fault. This chapter also describes the “notifier” component, consisting of the VCS notifier process and hanotify utility.

Chapter 11, “[Event Triggers](#),” describes how event triggers work and how they enable the administrator to take specific actions in response to particular events. This chapter also includes a description of each event trigger, including usage and location.

Chapter 12, “[VCS Performance Considerations](#),” describes the impact of VCS on overall system performance, and includes methods on how to adjust VCS to meet specific configuration requirements.

Chapter 13, “[Troubleshooting and Recovery](#),” explains VCS logging and defines the various message tags associated with log files. This chapter also describes how VCS handles network failure and how to troubleshoot common problems in Cluster Manager, VCIM, and bundled agents. The procedure for disabling resources while keeping a service group online is also presented in this chapter.

Appendix A, “[System States](#),” describes the various system states and the order in which a system transitions from one state to another.

Appendix B, “[VCS Attributes](#),” lists the VCS attributes for each cluster object, including service groups, resources, resource types, systems, clusters.

Appendix C, “[Additional Administration Matrices](#),” describes user privileges for less frequently used operations.

## Technical Support

For assistance with any VERITAS product, contact Technical Support:

U.S. and Canada: call 1-800-342-0652.

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

Software updates, TechNotes, product alerts, and hardware compatibility lists are also available from <http://support.veritas.com>.

To learn more about VERITAS and its products, visit <http://www.veritas.com>.

## Conventions

Typeface/Font	Usage
<b>bold</b>	names of screens, windows, tabs, dialog boxes, options, buttons
<i>italic</i>	new terms, book titles, emphasis, variables in tables or body text
Courier	computer output, command references within text
<b>Courier</b> (bold)	command-line user input, keywords in grammar syntax
<b><i>Courier</i></b> (bold, italic)	variables in a command
Symbol	Usage
#	superuser prompt (for all shells)







# Section I Overview of Cluster Technology

---

This section presents an overview of clustering software, including current industry trends and best practices, and describes the “building blocks” of VCS and basic configuration concepts. This information lays the groundwork for a solid understanding of cluster technology.

Section I includes the following chapters:

- ◆ **Chapter 1 “Getting Acquainted with Clustering” 1**
- ◆ **Chapter 2 “What is VERITAS Cluster Server?” 11**

# Getting Acquainted with Clustering

---

1

VERITAS® Cluster Server™ (VCS) provides a framework for application management and availability that extends far beyond simple hardware failure recovery. This chapter describes the basic concepts and terminology of application clustering, and introduces features unique to VCS.

## What Clustering Offers

### Increased Failover Granularity

In first-generation systems, granularity was at the server level, meaning if a high-availability application failed on a system, all applications failed over to a second system. This method severely limited server scalability. For example, running multiple production Oracle instances on a single system was problematic because the failure of one instance on the system brought down all instances while applications were migrated to another server.

As customers move into larger “enterprise class” servers, rarely is an entire server dedicated to a single application service, also known as a *service group*. Failures affecting a service group, such as a software or resource failure, should not affect other service groups residing on the same physical host. If they do, additional downtime is incurred.

Service groups address the issue of failover granularity. A service group is a collection of resources working together to provide application services to clients. It typically includes multiple resources, hardware- and software-based, working together to produce a single service. For example, a database service may comprise a logical network (IP) address, the DBMS software, underlying file systems, logical volumes, and a set of physical disks managed by VERITAS Volume Manager™ (VxVM). If this service group migrates to another node for recovery purposes, all of its resources must migrate together to re-create the group on another node, without affecting other service groups.



A single, large node may host several service groups, each providing a discrete service to networked clients. If multiple service groups are running on a single node, they must be monitored and managed independently. Independent management enables a service group to be recovered automatically or idled manually (for administrative or maintenance reasons) without affecting other service groups running on the node. Of course, if the entire server crashes (as opposed to just a software failure or hang), all service groups on that node must be recovered elsewhere.

Service groups can be manipulated independently, so a failed node’s workload can be *load-balanced* across remaining cluster nodes and failed over successively (due to consecutive failures) without manual intervention, as shown in “N-to-N” on page 10. Service groups are further described on [page 18](#).

## Enhanced Storage Architecture

Scalability is a major benefit gained from today’s clustering software. Current high-availability products scale to eight or more nodes, and Storage Area Network (SAN) technology makes much larger clusters possible. VCS supports up to 32 nodes in a single cluster.

## Can My Application be “Clustered”?

Nearly all applications can be placed under cluster control provided the application meets specific criteria, including:

- ◆ Defined start, stop, and monitor procedures.
- ◆ Capability of storing required data on shared disks.
- ◆ Ability of application to be restarted in a known state.
- ◆ Adherence to license requirements and host name dependencies.

## Criteria for Start, Stop, and Monitor Procedures

***The application must have a defined procedure for starting.*** There should be a well-known set of steps or specific commands that can be used to start the application, plus any external requirements, such as mounted file systems, IP addresses, etc. For example, to control an Oracle database, VCS requires the Oracle user, Instance ID, Oracle home directory, and the pfile. The developer must also know implicitly which disk groups, volumes and file systems are required.

***The application must have a defined procedure for stopping.*** Clusters running multiple instances of an application must have a method for stopping a single instance without affecting the others. For example, killing all HTTPd processes on a Web server is unacceptable because it would also stop other Web servers. In the case of Apache, the documented process for shutdown involves locating the PID file written by the specific instance on startup, and sending the PID a `kill -TERM` signal. This causes the master HTTPd process for that particular instance to halt all child processes.

If the cluster package is incapable of stopping an application cleanly, it may call a more forceful method. For example, if an Oracle database hangs and will not respond to `shutdown immediate`, the cluster package may call `shutdown abort`, which may in turn require identifying and releasing shared memory segments or semaphores. The need for a clean capability should be evaluated for each application and, when required, a method to clean up after an application must be identified.

In addition to a procedure for stopping an application, the need for a `clean` capability should be evaluated for each application and, when required, a method to clean up after an application must be identified.

***The application must have a defined procedure for monitoring the overall “health” of an individual instance.*** Using the Web server as an example, simply checking the process table for the existence of “httpd” is unacceptable, as any Web server would cause the monitor to return an online value. Checking if the PID contained in the PID file is actually in the process table is a better solution. Taking this check one step further, verify the process in the proc table is the correct HTTPd process to ensure the operating system has not reused the PID.

To add more robust monitoring, an application can be monitored closer to the user perspective. For example, an HTTPd server can be monitored by connecting to the correct IP address and port and testing if the Web server responds to `http` commands. In a database environment, the monitoring application can connect to the database server and perform SQL commands and verify read and write to the database. In both cases, end-to-end monitoring is a far more robust check of application health. The closer a test comes to matching what a user actually does, the better the test is in discovering problems. However there is a tradeoff: end-to-end monitoring increases system load and may increase system response time. From a design perspective, the level of monitoring should be a careful balance between ensuring the application is up and minimizing monitor overhead.



## External Data Storage

The application must be capable of storing all required data on shared disks. This may require specific setup options or soft links. For example, a product may only install in `/usr/local`. This would require linking `/usr/local` to a file system mounted from the shared storage device or actually mounting file system from the shared device on `/usr/local`. The application must also store data to disk rather than maintaining it in memory.

The takeover system must be capable of accessing all required information. This precludes the use of anything inside a single system inaccessible by the peer, such as NVRAM accelerator boards and other disk-caching mechanisms contained in a local host. Disk caching for performance is acceptable, but it must be done on the external array and not on the local host.

## Restarting the Application to a Known State

The application must be able to restart in a known state. This is probably the most important application requirement. On a switchover, the application is brought down under controlled conditions and started on another node. The requirements here are straightforward. The application must close out all tasks, store data properly on shared disk, and exit. At this time, the peer system can start up from a clean state. The problem arises when one server crashes and another must take over. The application must be written in such a way that data is not stored in memory but regularly written to disk.

A commercial database such as Oracle is the perfect example of a well-written, crash-tolerant application. On any client SQL request, the client is responsible for holding the request until it receives acknowledgement from the server. When the server receives a request, it is placed in a special log file, or “redo” file. The data is confirmed as being written to stable disk storage before acknowledging the client. Later, Oracle moves the data from redo log to actual table space. This is known as “checkpointing.” After a server crash, Oracle recovers the last known committed state by mounting the data tables and applying the redo logs. This in effect returns the database to the exact time of the crash. The client resubmits any outstanding client requests unacknowledged by the server, and all others are contained in the redo logs. An important point to note is the cooperation between the client application and the server. This must be factored in when assessing whether the application is cluster-compatible.

An important point to note is the cooperation between the client application and the server. This too must be factored in when assessing whether the application is cluster-compatible.

---

## Licensing and Host Name Issues

The application must be capable of running on all servers designated as potential hosts, which means strict adherence to licensing requirements and host name dependencies. Many customers have used custom scripting within cluster packages to modify a system host name on failover, but this is *not* recommended. Changing host names can lead to significant management issues when multiple systems have the same host name after an outage. It is better to configure applications and licensing to run properly on all hosts. Also, for best results, make sure to test-run your applications on all hosts prior to making it part of the cluster.

## Detecting Failure

### Application Failure

VCS can detect failure of an application or node. When VCS detects the failure of a configured application or resource, it can restart the application locally, or move the application and its associated resources to another node in the cluster.

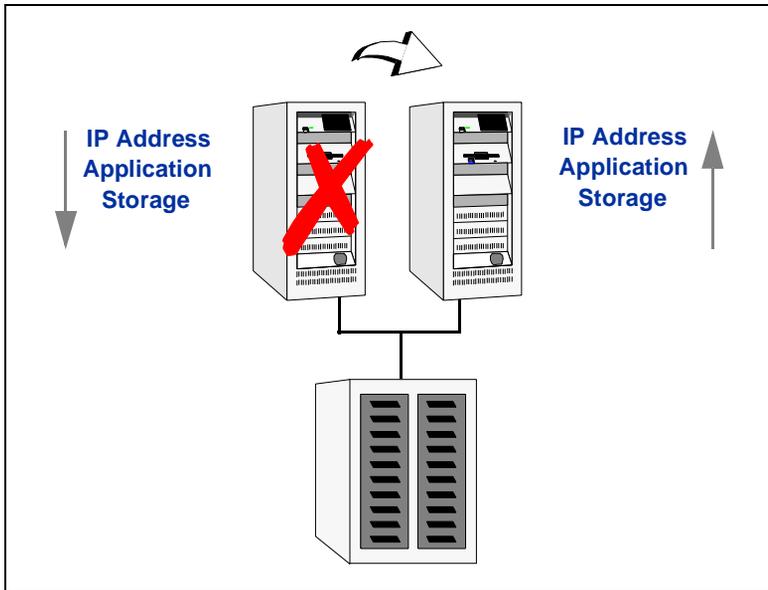
### Node Failure

*Node failure* means a node in the cluster has gone down or is not responding. One of the most difficult tasks in a cluster package is distinguishing between a loss of an actual node and a loss of communication between nodes. Several technologies are used to monitor node status and communication, including heartbeat networks between servers, quorum disks, SCSI reservation, and others. The difficulty is providing a reliable mechanism that scales for multiple nodes. VCS uses a robust network heartbeat mechanism, with optional heartbeat over shared disk.



## Failover and Switchover

Cluster products automate the process of bringing applications online on a spare, or “failover,” node. From a high level, this requires removing the IP address known to the clients, stopping necessary applications, and unmounting file systems from the failed node. On the spare node the process is then reversed: file systems are mounted, applications are started, and the IP address known to the clients is configured.

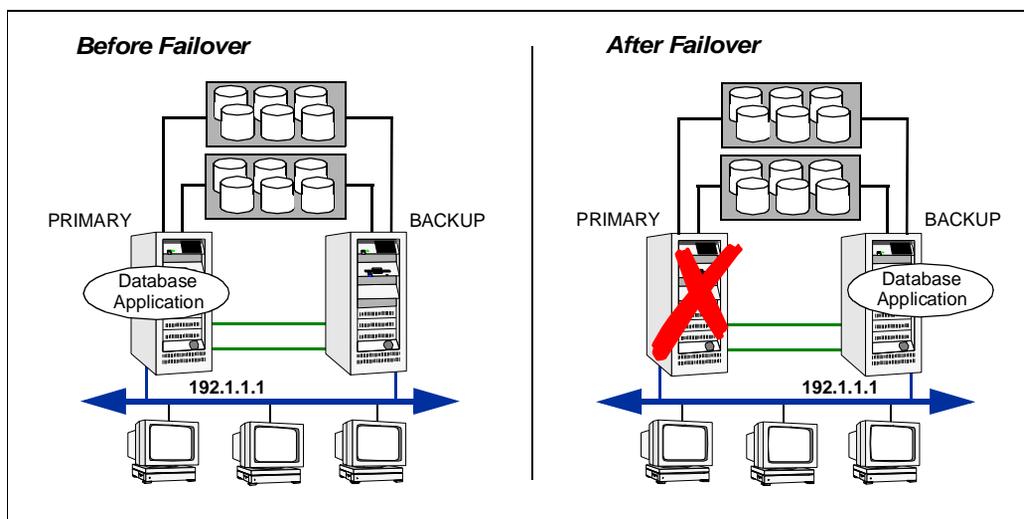


An operator-requested move between systems is a *switchover*. Switching applications between servers enables server maintenance and upgrades with little or no impact on end-users.

# Failover Configurations

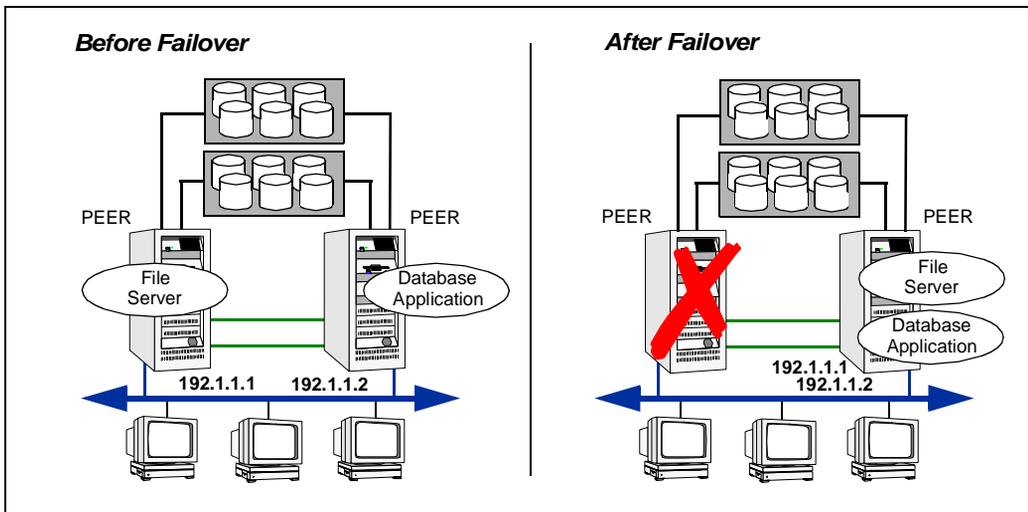
## Asymmetric

In an asymmetric configuration, an application runs on a primary server. A dedicated secondary, or “backup,” server is present to take over when the primary fails. The backup server is passive, meaning it is not configured to perform any other functions. In the following illustration, a database application is moved, or “failed over,” from the primary to the backup. Notice the IP address used by the clients moves as well. This is extremely important: If IP addresses were not moved, all clients would have to be updated on each server failover.



## Symmetric

In a symmetric configuration, each server is configured to run a specific application or service and provide backup for its peer. In the example below, the file server fails and its peer takes on both roles. Notice the surviving peer server now has two IP addresses assigned.



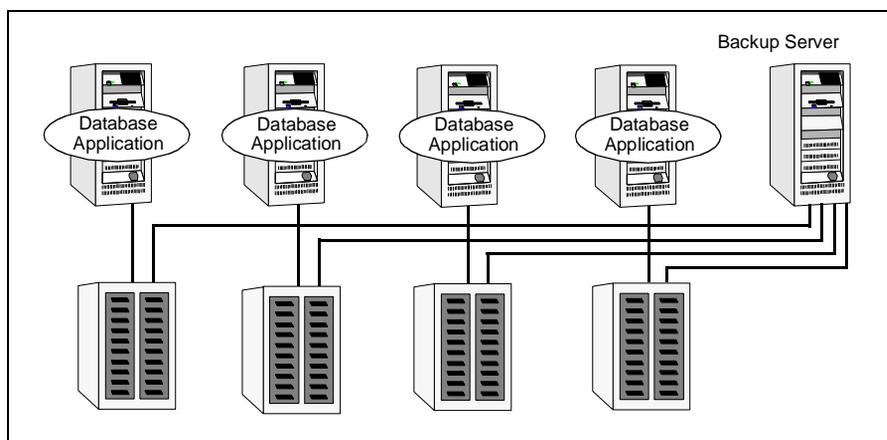
Generally speaking, it appears that symmetric configuration is far more efficient in terms of hardware utilization. Many customers object to the concept of a valuable system sitting idle. However, this line of reasoning can be flawed. In the asymmetric example, the backup server requires only as much processor power as its peer. On failover, performance remains the same.

In the symmetric example, the backup server requires not only enough processor power to run the existing application, but also enough to run the new application it takes over. To put it another way, if a single application requires one processor to run properly, an asymmetric configuration requires two, single-processor systems. However, to run identical applications on each server, a symmetric configuration requires two, dual-processor systems.

Further issues can arise in symmetric configurations when multiple applications running on the same system do not co-exist properly. Some applications work well with multiple copies started on the same system, but others fail. Even more difficult is two applications with different I/O and memory requirements running on the same system. Thorough testing in a symmetric configuration can uncover most issues. Today, many clusters employ symmetric configurations. When a server fails in this configuration, performance level remains acceptable for the short time it takes to restore the server.

## N-to-1

As mentioned previously, in an asymmetric configuration there is no performance penalty and no issues with multiple applications running on the same system; however, the drawback is the 100 percent redundancy cost at the server level. An N-to-1 configuration reduces hardware redundancy cost and provides a dedicated backup server.



N-to-1 Failover Configuration

An N-to-1 configuration is based on the concept that multiple, simultaneous server failures are unlikely; therefore, a single backup server can protect multiple active servers. When a server fails, its applications move to the backup server. For example, in a 4-to-1 configuration, one server can protect four servers, which reduces redundancy cost at the server level from 100 percent to 25 percent.

Nearly all cluster products can be configured as N-to-1, but storage considerations can make it impractical. The number of I/O slots in the backup server limits the number of servers in the cluster. Attaching more than two hosts to a single SCSI storage device is a problem because specialized cabling is required and it severely limits the number of SCSI disks that can be placed on the bus. A key technology employed by more advanced clusters is the Storage Area Network (SAN). SANs provide the ability to connect many hosts to an essentially unlimited amount of storage, allowing large clusters to be constructed easily.



## Advanced Failover Configurations

The advent of SANs, combined with second-generation high-availability (HA) products such as VCS, has enabled several new and useful failover configurations, described in the following sections.

### N + 1

SANs enable the creation of large clusters by connecting multiple servers to the same storage, which eliminates the need for a backup server. Instead of configuring clusters as N-to-1, users now have the option of configuring them as  $N + 1$ .

In advanced N + 1 configurations, an extra, or “spare,” server in the cluster provides additional horsepower only. When a primary server fails, the application restarts on the spare. When the original, primary server is repaired, it then becomes the spare server. This configuration eliminates the need for a second application failure to fail back the service group to the primary system. Any server can act as the spare to any other server. This allows clusters of eight or more nodes to use a single spare server. Cascading failover can also accommodate multiple server failures; however, this requires thorough testing and planning.

### N-to-N

N-to-N clustering is at the core of HA architecture supporting multiple applications. N-to-N refers to multiple service groups running on multiple servers, with each service group capable of being failed over to different servers in the cluster. For example, consider a four-node cluster with each node supporting three critical database instances. If any node fails, each instance is started on a different node, ensuring no single node becomes overloaded. This configuration is a logical evolution of N + 1: it provides for the cluster standby *capacity* instead of a standby *server*.

In an N-to-N configuration, cascading failover is also possible. The benefit of an N-to-N configuration is early testing of application inoperability. Configuring cascading failover is simply a matter of determining additional load each server can handle.



## The Building Blocks of VCS

VERITAS Cluster Server (VCS) enables you to monitor systems and application services, and to restart services on a different system when hardware or software fails. The key to understanding VCS is to learn the various “building blocks” of the product and how they interact with one another. They include:

- ◆ clusters
- ◆ resources and resource types
- ◆ service groups
- ◆ agents
- ◆ communications (GAB and LLT)

## Clusters

A VCS cluster consists of multiple systems connected in various combinations to shared storage devices. All systems within a cluster have the same cluster ID, and are connected by redundant private networks over which they communicate by *heartbeats*, signals sent periodically from one system to another. Within a single VCS cluster, all nodes must run on the same operating system.

Clusters can have from 1 to 32 member systems, or “nodes.” (The terms are used interchangeably throughout this guide.) Applications can be configured to run on specific nodes within the cluster. Storage is configured to provide access to shared data for nodes hosting the application, so storage connectivity determines where applications are run. All nodes sharing access to storage are eligible to run an application. Nodes that do not share storage cannot fail over an application that stores its data on disk.

Multiple clusters can be managed from the command-line interface and from two cross-platform graphical user interfaces: Cluster Manager (Java Console) and Cluster Manager (Web Console).



## Resources and Resource Types

Resources are hardware or software entities, such as network interface cards (NICs), IP addresses, applications, and databases, that are brought online, taken offline, or monitored by VCS. Each resource is identified by a unique name. Resources with similar characteristics are known collectively as a *resource type*; for example, two IP resources are both classified as type IP. How VCS starts and stops a resource is specific to the resource type. An IP resource is started by assigning the IP address to a NIC.

Monitoring a resource means testing it to determine if it is online or offline. How VCS monitors a resource is also specific to the resource type. Continuing with the example above, an IP address tests as online if it is present in the IP address of the NIC.

## Resource Categories

There are three categories of resources in VCS: *On-Off*, *On-Only*, and *Persistent*. Most resources are On-Off, meaning VCS starts and stops them as required. For example, VCS assigns the IP address to the specified NIC and removes the assigned IP when the associated service group is taken offline.

Other resources may also be required by VCS and external applications; for example, NFS daemons. VCS requires NFS daemons to be running to export a file system. There may also be other file systems exported locally, outside VCS control. The NFS resource is an On-Only resource, meaning that VCS starts the daemons if required, but does not stop them if the associated service group is taken offline. An On-Only resource is brought online when required by VCS, but is not taken offline when the associated service group is taken offline.

A Persistent resource cannot be brought online or taken offline, yet VCS requires the resource to be present in the configuration. For example, a NIC cannot be started or stopped, but it is required to configure an IP address. VCS monitors Persistent resources to ensure their status and operation.



## Service Groups

As mentioned in [Chapter 1](#), service groups mark the primary difference between first- and second-generation high-availability (HA) packages. Early systems used the entire server as the granularity level for failover: if an application failed, all applications were migrated to a second system. Second-generation HA packages, such as VCS, greatly increase the granularity of application control. This smaller container for applications and associated resources is called a *service group*.

For example, a service group for a Web application may consist of:

- ◆ disk groups on which Web pages are stored
- ◆ a volume built in the disk group
- ◆ a file system using the volume
- ◆ a database whose table spaces are files and whose rows contain page pointers
- ◆ network interface cards to export the Web service
- ◆ one or more IP addresses associated with the network cards
- ◆ the application program and associated code libraries

VCS performs administrative operations on resources at the service group level, including starting, stopping, restarting, and monitoring. For example, when a service group is brought online, all resources within the group are also brought online. When a failover occurs in VCS, resources never fail over individually: the entire service group containing the resource fails over as a unit. If there is more than one group defined on a server, one group may fail over without affecting the other groups.

Additionally:

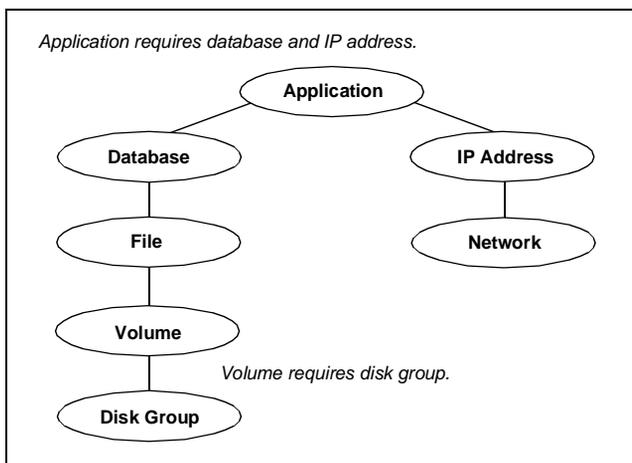
- ◆ If a service group is to run on a particular server, all of the group's required resources must be available to the server.
- ◆ Resources within a service group have *dependencies*; that is, some resources, such as volumes, must be operational before other resources, such as the file system, can become operational.



## Resource Dependencies

One of the most important concepts of the service group definition is *resource dependencies*. As mentioned in the preceding section, when a service group is brought online or taken offline, the resource dependencies within the group specify the order in which the resources are brought online and taken offline. For example, a VERITAS Volume Manager™ disk group must be imported before volumes in the disk group are started, and volumes must be started before file systems are mounted. Conversely, file systems must be unmounted before volumes are stopped, and volumes must be stopped before disk groups are deported.

In VCS terminology, resources are categorized as *parents* or *children*, depending on how they are configured. Diagramming the relationship between them forms a *graph*. Parent resources appear at the top of the “arcs” that connect them to their child resources. Typically, child resources are brought online before parent resources, and parent resources are taken offline before child resources. Resources must adhere to the established order of dependency. The dependency graph is an easy way to document resource dependencies within a service group. The following figure shows a resource dependency graph for a cluster service.



In the figure above, the lower (*child*) resources represent resources required by the upper (*parent*) resources: the volume requires that the disk group be online, the file system requires that the volume be active, and so on. The application requires two, independent resource “subtrees”: a database and an IP address for client communications.

VCS includes a language for specifying resource types and dependency relationships. The VCS engine, referred to throughout this guide as “HAD” (high-availability daemon) uses resource and dependency definitions when bringing applications online or taking them offline. Typically, child resources must be functioning before parent resources can be brought online. In the preceding figure, the disks and the network card may be brought online concurrently because they have no dependencies. When all child resources required by a parent are online, the parent itself is brought online, and so on up the graph, until finally the application is started.

Conversely, when taking a service offline, the VCS engine begins at the top of the graph. In the preceding example, the application would be taken offline first, followed by the database and the IP address in parallel, and so on down the graph until finally the application is taken offline.

## Types of Service Groups

In VCS there are two categories of service groups: *failover* and *parallel*. A failover group runs on one system in the cluster at a time. A parallel group runs concurrently on more than one system at a time.

### Failover Groups

Failover groups are used for many application services, such as most databases and NFS servers. HAD assures that a failover service group is online only, partially online, or in a state other than OFFLINE (such as attempting to go online or attempting to go offline) on only one system at a time.

### Parallel Groups

Parallel groups are used far less frequently than failover groups, and are more complex. They require applications that can be started safely on multiple systems, without threat of data corruption. They also require that applications running multiple instances allow all instances access to the same data.



## Autostarting Resources

A service group's resources are brought online according to how the group was started: by command or at system startup. If the group was brought online by command, resources whose attributes are set to start automatically (the default setting) are brought online. However, before they can start, all of their child resources must first be brought online.

If the system is a member of a failover service group's `AutoStartList` attribute and the group is not already running on another system in the cluster, the group is brought online when the system is started. Resources and their dependencies set to start automatically are brought online after their child resources. If the system is a member of a parallel service group's `AutoStartList` attribute, the group is brought online when the system is booted.

## Agents

An agent is a program designed to control resources of a particular resource type. VCS includes a set of predefined resource types, and each has a corresponding *agent*, designed to control the resource. There is one agent for each resource type. Agents control resources according to information hardcoded into the agent itself, or by running scripts. Agents act as the “intermediary” between a resource and VCS. The agent recognizes the resource requirements and communicates them to VCS. For example, for VCS to bring an Oracle resource online it does not need to understand Oracle; it simply passes the online command to the Oracle agent, which, in turn, knows to call the server manager and issue the appropriate startup command.

Agents can try to restart a failed resource prior to declaring it faulted. A resource cannot be brought online or taken offline without an agent, and the actions required to do either differ significantly from resource to resource. VCS handles the difference in a way that makes it simple for application and hardware developers to integrate additional resources into the cluster framework. VCS agents are multithreaded, meaning a single VCS agent monitors multiple resources of the same resource type on one host. For example, the Disk agent monitors all disk resources.

VCS monitors resources when they are online *and* offline to ensure they are not started on systems on which they are not supposed to run. For this reason, when VCS is started on a system, it starts the agent for any resource configured to run on that system. If there are no resources of a particular type configured to run on the system, the agent is not started.

VCS agents are located in the `/opt/VRTSvcs/bin` directory. For example, the Disk agent and the corresponding online, offline, and monitor scripts are located in the directory `/opt/VRTSvcs/bin/Disk`.



## Agent Classifications

There are three classifications of VCS agents:

- ◆ *Bundled*

Bundled agents are packaged with VCS. They include agents for Disk, Mount, IP, and various other resource types. See the *VERITAS Bundled Agents Reference Guide* for a complete list.

- ◆ *Enterprise*

Enterprise agents are packaged separately and sold by VERITAS to control third-party applications. They include agents for Informix, Oracle, NetBackup, and Sybase, and others. Each enterprise agent includes instructions on installing and configuring the agent. Contact your VERITAS sales representative for more information.

- ◆ *Custom*

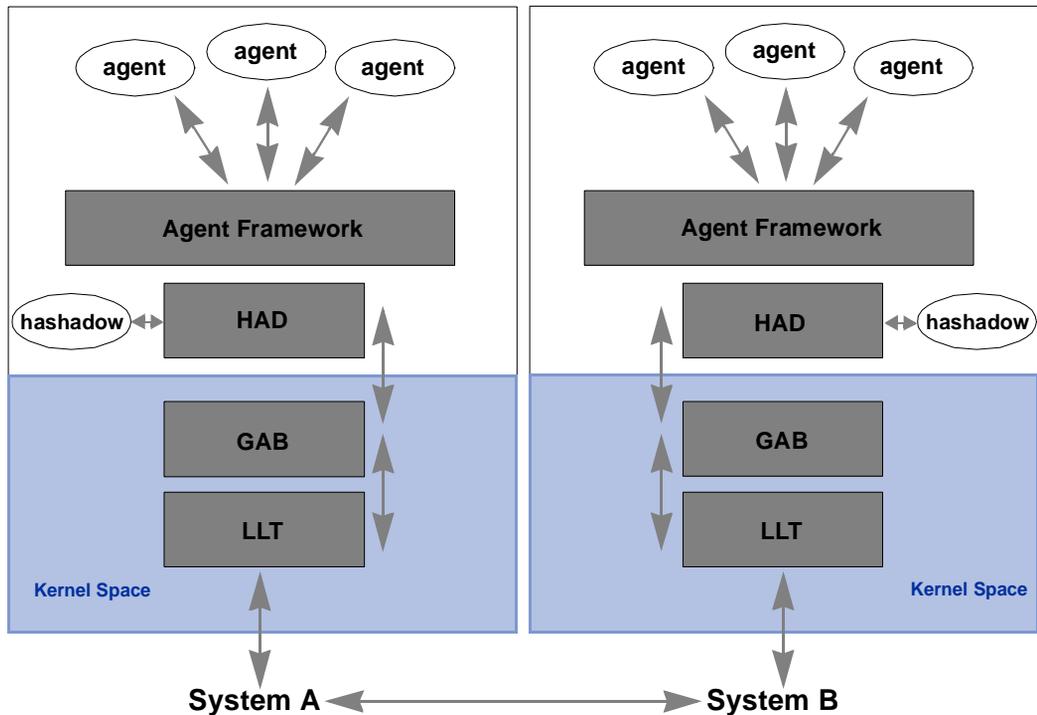
Custom agents can be developed by the user or by VERITAS consultants. Typically, agents are developed because the user requires control of an application that is not covered by current bundled or enterprise agents. See the *VERITAS Cluster Server Agent Developer's Guide* for information on developing your own custom agent, or contact VERITAS Enterprise Consulting Services.



## Cluster Communications

Cluster communications ensure VCS is continuously aware of the status of each system's service groups and resources. They also enable VCS to recognize which systems are active members of the cluster, which are joining or leaving the cluster, and which have failed.

The following figure presents an overview of VCS communications. On each cluster system, agents monitor the status of resources and communicate the status to the high-availability daemon, "HAD." HAD then communicates the status on the local system to other systems in the cluster via the Group Membership Services/Atomic Broadcast protocol (GAB) and the Low Latency Transport (LLT). Descriptions of each of these components begin on [page 19](#).



## Key Components of Cluster Communications

As shown in the preceding figure, there are several components that make up cluster communications. The following subsections describe the role of each component, including network and disk channels, which are not shown in the figure.

### Group Membership Services/Atomic Broadcast (GAB)

GAB is the mechanism for monitoring cluster memberships, tracking cluster state, and distributing the information to cluster systems. In VCS, *cluster membership* is defined as all systems configured with the same cluster ID interconnected via a pair of redundant heartbeat networks. During standard operation, all systems configured as part of the physical cluster during system installation are actively participating in cluster communications. Cluster membership enables VCS to dynamically track the entire cluster topology.

Systems join a cluster by issuing a “Cluster Join” message during GAB startup. Cluster membership is maintained by the use of *heartbeats*, signals sent periodically from one system to another to verify the systems are active. Heartbeats over the network are handled by the LLT protocol (described in the following section). When systems stop receiving heartbeats from a peer for the interval specified in the Heartbeat Timeout attribute, the system in question is marked `DOWN` and excluded from the cluster. Its applications are then migrated to the other systems.

*Cluster state* refers to tracking the status of all resources and groups in the cluster. This is the function of the atomic broadcast capability of GAB. Atomic broadcast ensures all systems within the cluster are immediately notified of changes in resource status, cluster membership, and configuration. *Atomic* means all systems receive updates, or are “rolled back” to the previous state, much like a database atomic commit. If a failure occurs while transmitting status changes, GAB’s atomicity ensures that upon recovery, all systems have the same information regarding the status of any monitored resource in the cluster. The broadcast messaging employs a two-phase commit protocol to deliver messages atomically to surviving members of a group.



## HAD

The high-availability daemon, HAD, is the primary process running on each system and is sometimes referred to as the “VCS engine.” It receives information from various agents regarding resources on the local system and forwards the information to each member system. It also receives information from other cluster members, which it uses to update its own “view” of the cluster. HAD is monitored and, when required, restarted by a process called “hashadow,” which also runs on each system in the cluster.

## Low Latency Transport (LLT)

LLT is the mechanism that provides communications between systems in a cluster. Specifically, it provides fast, kernel-to-kernel communications, and monitors network connections. It serves as a replacement for the IP stack on systems, and runs directly on top of the Data Link Protocol Interface (DLPI) layer on UNIX, and the Network Driver Interface Specification (NDIS) on Windows. Using LLT rather than IP removes latency and overhead associated with the IP stack, and ensures that events such as state changes are reflected more quickly.

LLT distributes (“load-balances”) internode communication across private network links. This means cluster state information is evenly distributed across all private network links (maximum eight) to ensure performance and fault resilience. When a link fails, traffic is redirected to remaining links.

LLT is also responsible for sending and receiving heartbeat traffic over network links. The frequency of heartbeats can be set in the file `/etc/llttab`. Heartbeats are used to determine the “health” of nodes in the cluster.

LLT also informs GAB if communications to a peer are *reliable* or *unreliable*. A peer connection is reliable if more than one network link exists between them. LLT monitors multiple links and routes network traffic over the surviving links.

For example, if two fully connected, independent networks exist between Systems A, B and C, and if one network interface card on System C fails, LLT on Systems A and B routes traffic to System C over the remaining interface card while multiplexing traffic to Systems A and B over both networks. The result is Systems A and B have a reliable connection with each other, and an unreliable connection to System C, and vice versa: System C has an unreliable connection to Systems A and B.

For reliable communication to work, it is critical the networks fail independently. LLT supports multiple independent links between systems. Using different interfaces and connecting infrastructure reduces the risk of simultaneous link failure and increases overall reliability.

## LLT's Low Priority Link

LLT can be configured to use a low-priority network link as a backup to standard heartbeat channels. Low-priority links are typically configured on the customer's public network or administrative network. The low-priority link is not used for cluster membership traffic until it is the only remaining link.

During standard operation, the low-priority link carries heartbeat traffic for cluster membership and link state maintenance only. The frequency of heartbeats decreases 50 percent to reduce network overhead. When the low-priority link is the sole network link remaining, LLT switches all cluster status traffic to the low-priority link. Upon repair of any configured private link, LLT returns cluster status traffic to the high-priority link.

## Network and Disk Channels

Though both network and disk channels are used for VCS communications, there are important distinctions between the two:

- ◆ Both can be used for cluster membership.

Cluster membership requires a simple, unacknowledged broadcast to all nodes in the cluster that basically says "I am System X and I am here." This is the basic function of the heartbeat mechanism, and is relatively lightweight when compared to the requirements for monitoring cluster state.

- ◆ Only network channels can be used to transmit cluster state.

Monitoring cluster state requires considerably more information to be passed between nodes than cluster membership. Disk channels *cannot* transmit cluster state.

Network and disk channels, and their role in handling network failure, are described further on [page 311](#).

## Disk Heartbeats (GABDISK)

Another way to improve cluster resiliency is a disk heartbeat, or "GABDISK," which allows a heartbeat to be placed on a physical disk shared by all systems in the cluster. A disk heartbeat uses two small, dedicated regions of a physical disk, and has the following limitations:

- ◆ Cluster size is limited to a maximum of eight nodes.
- ◆ For optimum performance disk heartbeats require a dedicated disk. GABDISK is "chatty" and will adversely affect the performance of applications accessing the disk. At the same time, heavy access by an application to a disk used for GAB may cause heartbeat timeouts.

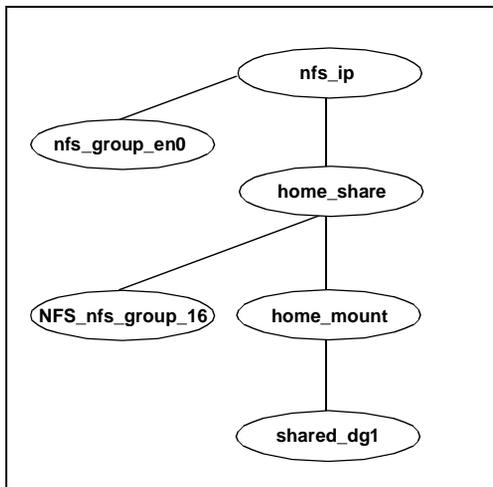


## Fitting the Blocks Together

Understanding how VCS building blocks fit together is the key to understanding VCS as a whole. In the following example, a two-node cluster is sharing directories to clients. Both nodes are connected to shared storage, which allows them access to the directories being shared. In this example, a single service group is configured, “FileShare\_Group,” that will be failed over between System A and System B, as necessary. The service group, configured as a failover group, consists of various resources, each with a different resource type. As described in “[Resource Dependencies](#),” the resources must be started in a specific order.

To control each resource type, VCS requires an agent. The VCS engine, “HAD,” reads the configuration file and determines what agents are required to control the resources in this group (plus resources in other service groups configured to run on the system) and starts the corresponding VCS agents. HAD then determines the order in which to bring the resources online, based on the resource dependency statements in the configuration. When it is time to bring the service group online, VCS issues online commands to the corresponding agents in the correct order.

The following figure shows the dependency graph for the service group NFS\_Group.



In this configuration, the VCS engine will start agents for disk group, mount, share, NFS, NIC, and IP on all systems configured to run NFS\_Group. The resource dependencies are configured as:

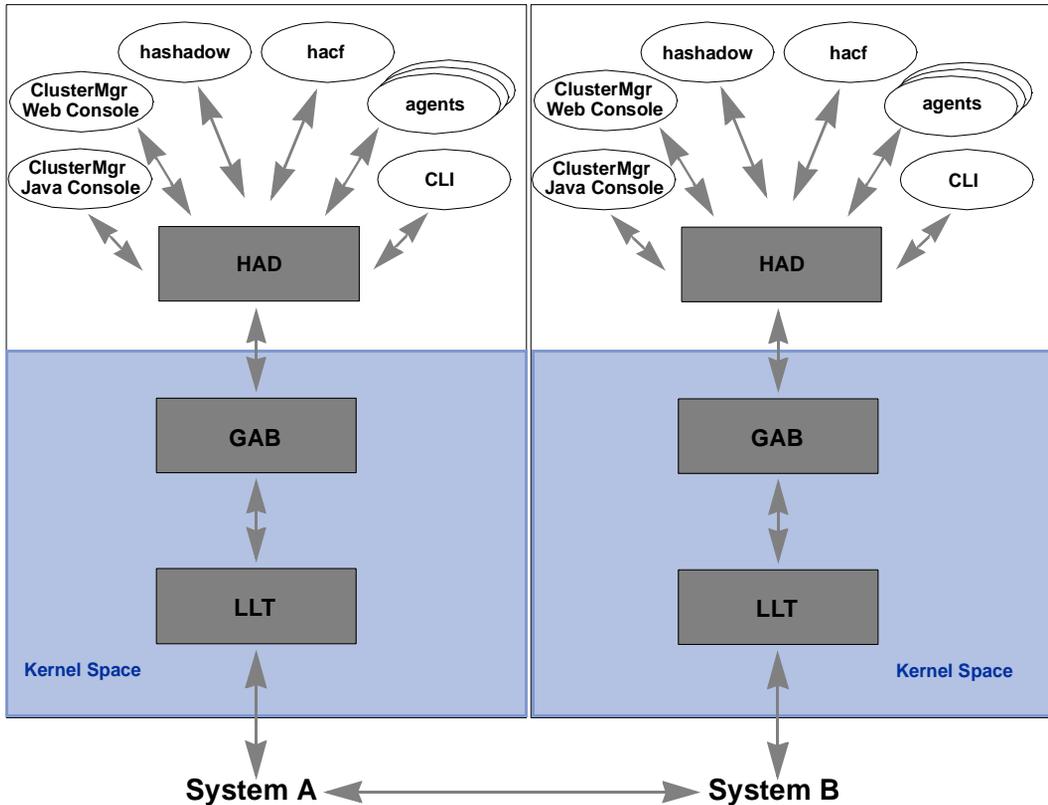
- ◆ The /home file system, home\_mount, requires the disk group, shared\_dg1, to be online before mounting.
- ◆ The NFS export of the /home file system requires the /home file system be mounted and the NFS daemons be running.
- ◆ The high-availability IP address, nfs\_IP, requires the file system to be shared and the network interface to be up, represented as nfs\_group\_en0.
- ◆ The NFS daemons and the disk group have no lower (child) dependencies, so they can start in parallel.
- ◆ The NIC resource is a persistent resource and does not require starting.

The service group NFS\_Group can be configured to start automatically on either node in the preceding example. It can then move or fail over to the second node on command or automatically if the first node fails. Upon failover or relocation, VCS will offline the resources beginning at the top of the graph and start them on the second node beginning at the bottom.



## VCS Processes

In addition to the processes and components previously cited in this chapter, there are several others that play a key role in VCS operations. The following figure shows the processes that run on each system in a cluster. Several of these were described previously beginning on [page 19](#), including, HAD, GAB, LLT, and agents. The remaining processes are described beginning on [page 25](#).



---

## Command-Line Interface (CLI)

The VCS command-line interface provides a comprehensive set of commands for managing and administering the cluster. Use as an alternative to, or in conjunction with, the two VCS graphical user interfaces: Cluster Manager (Java Console) and Cluster Manager (Web Console).

## Cluster Manager (Java Console)

A cross-platform, Java-based graphical user interface comprising “Cluster Manager” and “Configuration Editor” that provides complete administration capabilities for your cluster. It runs on any system inside or outside the cluster and on any operating system that supports Java.

## Cluster Manager (Web Console)

A cross-platform, Web-based graphical user interface for monitoring the cluster and performing basic cluster administration, including many of the same operations as the command-line interface and Cluster Manager (Java Console).

## The hacf Utility

This utility is executed on demand. It is used to verify a configuration file and can also be used by HAD to load a configuration file at run time.

## The hashadow Process

A process that monitors and, when required, restarts HAD.







## Section II VCS Configuration–The Basics

---

This section describes the VCS configuration language and basic configuration concepts, such as the main configuration files and how to modify and maintain a configuration. It also defines the concept of a service group dependency, and describes the various categories, locations, and types of dependencies. We recommend reading this section prior to installing and configuring VCS.

Section II includes the following chapters:

- ◆ **Chapter 3 “Configuration Concepts” 29**
- ◆ **Chapter 4 “The Role of Service Group Dependencies” 43**

# Configuration Concepts

---

# 3

Configuring VCS means conveying to the VCS engine the definitions of the cluster, service groups, resources, and resource dependencies. VCS uses two configuration files in a default configuration:

- ◆ The `main.cf` file defines the entire cluster.
- ◆ The `types.cf` file defines the resource types.

By default, both files reside in the directory `/etc/VRTSvcs/conf/config`. Additional files similar to `types.cf` may be present if agents have been added, such as `Oracletypes.cf`.

In a VCS cluster, the first system to be brought online reads the configuration file and creates an internal (in-memory) representation of the configuration. Systems brought online after the first system derive their information from systems already running in the cluster. When a configuration is changed dynamically, or when a system joins a running cluster, the current running cluster configuration is written to the files `main.cf`, `types.cf`, and any files included by `main.cf`. This is important to note when creating or modifying configuration files. Note that the cluster must be stopped while you are modifying the files. Changes made by editing the configuration files directly take effect when the cluster is restarted. The node on which the changes were made should be the first node brought back online.



## The VCS Configuration Language

The VCS configuration language specifies the makeup of service groups and their associated entities, such as resource types, resources, and attributes. These specifications are expressed in configuration files, the names of which contain the suffix `.cf`.

For example, the body of the configuration is in `main.cf`. Using an include statement, it references the file `types.cf`, which specifies resource types.

There are three ways to generate configuration files:

- ✓ Use the Java Console (Cluster Manager or Configuration Editor). Cluster Manager is the graphical user interface for VCS. Configuration Editor is a configuration tool that enables you to generate new configuration files (`main.cf` and `types.cf`) while VCS is offline. (See “[Administering VCS from Cluster Manager \(Java Console\)](#)” for details.)
- ✓ Use the command-line interface to modify the configuration.
- ✓ If VCS is not running, use a text editor to create and modify the files.

## The `main.cf` File

The format of the `main.cf` file comprises include clauses and definitions for the cluster, systems, service groups, and resources. The `main.cf` file also includes service group and resource dependency clauses.

### Include Clauses

Include clauses incorporate additional configuration files into `main.cf`. These additional files typically contain type definitions. At minimum, the `types.cf` file is included. Other type definitions must be configured as required. Typically, VCS enterprise agents add type definitions in their own files, as do custom agents developed for the cluster. Most customers and VERITAS consultants do not modify the `types.cf` file, but instead create additional type files.

### Cluster Definition

This section of `main.cf` defines the attributes of the cluster, including the cluster name and the names of the cluster GUI users.



## System Definition

Each system designated as part of the cluster is listed in this section of main.cf. The names listed as system names must match the name returned by the command `uname -a`. (If they do not match, see the *VERITAS Cluster Server Release Notes* for instructions on resolving the names.) System names are preceded with the keyword “system.” For any system to be used in a service group definition, it must be defined in this section. Consider this the overall set of available systems, and each service group a subset.

## Service Group Definition

Service group definitions in main.cf comprise the attributes of a particular service group. See “[Service Group Attributes](#)” on page 344 for a complete list. The following information describes two common service group attributes: `SystemList` and `AutoStartList`.

### SystemList Attribute

The `SystemList` attribute designates all systems that can run a particular service group. VCS does not allow a service group to be brought online on a system that is not in the group’s system list. By default, the order of systems in the list defines the priority of systems used in a failover. For example, the definition `SystemList = { SystemA, SystemB, SystemC }` configures SystemA to be the first choice on failover, followed by SystemB and then SystemC.

System priority may also be assigned explicitly in the `SystemList` attribute by assigning numeric values to each system name. For example: `SystemList = { SystemA=0, SystemB=1, SystemC=2 }` is identical to the preceding example, but in this case the administrator can change priority by changing the numeric values.

### AutoStartList Attribute

The `AutoStartList` attribute designates the system that brings up the service group on a full cluster start. If this system is not up when all others are brought online, the service group remains offline; for example, `AutoStartList = { SystemA }`.

## Resource Definition

This section in main.cf defines each resource used in a particular service group. Resources can be added in any order and the utility `hacf` arranges the resources alphabetically the first time the configuration file is run.



## Service Group Dependency Clause

To configure a service group dependency, place the keyword “requires” clause in the service group declaration of the main.cf file. Position the dependency clause before the resource dependency specifications and after the resource declarations.

## Resource Dependency Clause

A dependency between resources is indicated by the keyword “requires” between two resource names. This indicates the second resource (the child) must be online before the first resource (the parent) can be brought online. Conversely, the parent must be offline before the child can be taken offline. Also, faults of the children are propagated to the parent.

## Example 1: Initial Configuration

When VCS is installed, a basic main.cf configuration file is created with the cluster name, systems in the cluster, and a GUI user “admin” with the password “password.”

The following is an example of the main.cf for cluster “demo” and systems “SystemA” and “SystemB.”

```
include "types.cf"
cluster demo (
  UserNames = { admin = cDRpdxPmHzpS }
)
system SystemA
system SystemB
```

## Example 2: The main.cf for a Two-Node Asymmetric NFS Cluster

The following example is a basic two-node cluster exporting an NFS file system. The systems are configured as:

- ◆ servers: Server1 and Server2
- ◆ storage: One VxVM disk group, shared1
- ◆ file system: /home
- ◆ IP address: 192.168.1.3 IP\_nfs1
- ◆ public interface: en0
- ◆ Server1 is primary location to start the NFS\_group1

In an NFS configuration, the resource dependencies must be configured to bring up the IP address last. This prevents the client from accessing the server until everything is ready, and preventing unnecessary “Stale File Handle” errors on the clients.

```
include "types.cf"
cluster demo (
    UserNames = { admin = cDRpdxPmHpzS }
)

system Server1
system Server2

group NFS_group1 (
    SystemList = { Server1, Server2 }
    AutoStartList = { Server1 }
)

DiskGroup DG_shared1 (
    DiskGroup = shared1
)

IP IP_nfs1 (
    Device = en0
    Address = "192.168.1.3"
)

Mount Mount_home (
    MountPoint = "/export/home"
    BlockDevice = "/dev/vx/dsk/shared1/home_vol"
    FSType = vxfs
    MountOpt = rw
)
```



```
NFS NFS_group1_16 (  
    Nservers = 16  
)  
NIC NIC_group1_en0 (  
    Device = en0  
    NetworkType = ether  
)  
Share Share_home (  
    PathName = "/export/home"  
)  
  
IP_nfs1 requires Share_home  
IP_nfs1 requires NIC_group1_en0  
Mount_home requires DG_shared1  
Share_home requires NFS_group1_16  
Share_home requires Mount_home
```



## The types.cf File

The types.cf file describes standard resource types to the VCS engine; specifically, the data required to control a specific resource. The following example illustrates a DiskGroup resource type definition.

```
type DiskGroup (
    static int NumThreads = 1
    static int OnlineRetryLimit = 1
    static str ArgList[] = { DiskGroup, StartVolumes, StopVolumes,
        MonitorOnly }
    NameRule = resource.DiskGroup
    str DiskGroup
    str StartVolumes = 1
    str StopVolumes = 1
```

The types definition performs two important functions. First, it defines the type of values that may be set for each attribute. In the DiskGroup example, the NumThreads and OnlineRetryLimit attributes are both classified as `int`, or integer. The DiskGroup, StartVolumes and StopVolumes attributes are defined as `str`, or strings. See “[Attribute Data Types](#)” on page 37 for more information on integers and strings.

The second critical piece of information provided by the type definition is the ArgList attribute. The line `static str ArgList[] = { xxx, yyy, zzz }` defines the order parameters are passed to the agents for starting, stopping and monitoring resources. For example, when VCS wants to online the disk group “shared\_dg1”, it passes the online command to the DiskGroup agent with the following arguments:

```
shared_dg1 shared_dg1 1 1 <null>
```

The sequence of arguments indicates the online command, the name of the resource, then the contents of the ArgList. Since MonitorOnly is not set, it is passed as a null. This is always the order: command, resource name, ArgList.



For another example, review the following main.cf and types.cf representing an IP resource:

```
IP nfs_ip1 (  
    Device = en0  
    Address = "192.168.1.201"  
)  
  
type IP (  
    static str ArgList[] = { Device, Address, NetMask, Options }  
    NameRule = IP_ + resource.Address  
    str Device  
    str Address  
    str NetMask  
    str Options  
)
```

In this example, the high-availability address is configured on interface en0. The IP address is enclosed in double quotes because the string contains periods. See “[Attribute Data Types](#)” on page 37.

The VCS engine passes the identical arguments to the IP agent for online, offline, clean and monitor. It is up to the agent to use the arguments it requires. All resource names must be unique in a VCS cluster. If a name is not specified, the hacf utility generates a unique name based on the NameRule. The name generated by hacf in the example above is IP\_192.168.1.201.

## Attributes

VCS components are configured using *attributes*. Attributes contain data regarding the cluster, systems, service groups, resources, resource types, and agents. For example, the value of a service group's SystemList attribute specifies on which systems the group is configured and the priority of each system within the group. Each attribute has a definition and a value. You define an attribute by specifying its data type and dimension. Attributes also have default values that are assigned when a value is not specified.

### Attribute Data Types

Data Type	Description
String	A string is a sequence of characters enclosed by double quotes. A string may also contain double quotes, but the quotes must be immediately preceded by a backslash. A backslash is represented in a string as <code>\\</code> . Quotes are not required if a string begins with a letter, and contains only letters, numbers, dashes (-), and underscores (_). For example, a string defining a network interface such as <code>en0</code> does not require quotes as it contains only letters and numbers. However a string defining an IP address requires quotes, such as: <code>"192.168.100.1"</code> because the address contains periods.
Integer	Signed integer constants are a sequence of digits from 0 to 9. They may be preceded by a dash, and are interpreted in base 10. Integers cannot exceed the value of a 32-bit signed integer: 21471183247.
Boolean	A boolean is an integer, the possible values of which are 0 (false) and 1 (true).



## Attribute Dimensions

Dimension	Description
Scalar	A scalar has only one value. This is the default dimension.
Vector	A vector is an ordered list of values. Each value is indexed using a positive integer beginning with zero. A set of brackets ([]) denotes that the dimension is a vector. Brackets are specified after the attribute name on the attribute definition. For example, to designate a dependency between resource types specified in the service group list and all instances of the respective resource type:  <code>Dependencies = [ Mount, Disk, DiskGroup ]</code>
KeyList	A keylist is an unordered list of strings, and each string is unique within the list. For example, to designate the list of systems on which a service group will be started with VCS (usually at system boot):  <code>AutoStartList = { SystemA, SystemB, SystemC }</code>
Association	An association is an unordered list of name-value pairs. Each pair is separated by an equal sign. A set of braces ({} ) denotes that an attribute is an association. Braces are specified after the attribute name on the attribute definition. For example, to designate the list of systems on which the service group is configured to run and the system's priorities:  <code>SystemList = { SystemA=1, SystemB=2, SystemC=3 }</code>

## Type-Dependent Attributes

Type-dependent attributes pertain to a particular resource type. For example the `MountPath` attribute pertains only to the `Mount` resource type. Similarly, the `Address` attribute pertains only to the `IP` resource type.

## Type-Independent Attributes

Type-independent attributes apply to all resource types. This means there is a set of attributes that all agents can understand, regardless of resource type. These attributes are coded into the agent framework when the agent is developed. Attributes such as `RestartLimit` and `MonitorInterval` can be set for any resource type.



## Resource-Specific Attributes

Resource-specific attributes pertain to a specific resource only. They are discrete values that define the “personality” of a given resource. For example, the IP agent knows how to use the Address attribute. Setting an IP address is done only within a specific resource definition. Resource-specific attributes are set in the main.cf file.

## Type-Specific Attributes

Type-specific attributes are set for all resources of a specific type. For example, setting MonitorInterval for the IP resource affects all IP resources. The value for MonitorInterval would be placed in the types.cf file. In some cases, attributes can be placed in main.cf or types.cf. For example, setting StartVolumes = 1 in the DiskGroup types.cf entry would default StartVolumes to True for all DiskGroup resources. Placing the value in main.cf would set StartVolumes on a per-resource value.

In the example below, StartVolumes and StopVolumes is set in types.cf. This sets the default for all DiskGroup resources to automatically start all volumes contained in a disk group when the disk group is brought online. This is simply a default. If no value for StartVolumes or StopVolumes is set in main.cf, they will default to True.

```
type DiskGroup (
    static int NumThreads = 1
    static int OnlineRetryLimit = 1
    static str ArgList[] = { DiskGroup, StartVolumes, StopVolumes,
        MonitorOnly }
    NameRule = resource.DiskGroup
    str DiskGroup
    str StartVolumes = 1
    str StopVolumes = 1
)
```

Adding the required lines in main.cf allows this value to be overridden. In the next excerpt, the main.cf is used to override the default type-specific attribute with a resource-specific attribute:

```
DiskGroup shared_dg1 (
    DiskGroup = shared_dg1
    StartVolumes = 0
    StopVolumes = 0
)
```



In the next example, changing the `StartVolumes` and `StopVolumes` attributes to `static str` disables `main.cf` from overriding:

```
type DiskGroup (  
    static int NumThreads = 1  
    static int OnlineRetryLimit = 1  
    static str ArgList[] = { DiskGroup, StartVolumes, StopVolumes,  
        MonitorOnly }  
    NameRule = resource.DiskGroup  
    str DiskGroup  
    static str StartVolumes = 1  
    static str StopVolumes = 1
```

## Local and Global Attributes

An attribute whose value applies to all systems is *global* in scope. An attribute whose value applies on a per-system basis is *local* in scope. The “at” operator (@) indicates the system to which a local value applies. An example of local attributes can be found in the `MultiNICA` resource type where IP addresses and routing options are assigned per machine.

```
MultiNICA mnic (  
    Device@sysa = { en0 = "166.98.16.103", en3 = "166.98.16.103" }  
    Device@sysb = { en0 = "166.98.16.104", en3 = "166.98.16.104" }  
    NetMask = "255.255.255.0"  
    Options = "mtu m"  
    RouteOptions@sysa = "-net 192.100.201.0 192.100.13.7"  
    RouteOptions@sysb = "-net 192.100.201.1 192.100.13.8"  
)
```

## Keywords/Reserved Words

The following list includes the current keywords reserved for the VCS configuration language. Note they are case-sensitive.

action	firm	MonitorOnly	start
after	global	Name	Start
ArgList	group	NameRule	state
before	Group	Path	static
boolean	hard	Probed	stop
cluster	int	remote	str
Cluster	IState	requires	system
condition	keylist	resource	System
ConfidenceLevel	local	set	temp
event	offline	Signaled	type
false	online	soft	Type

## Managing the VCS Configuration File

### The hacf Utility

The hacf utility translates the VCS configuration language into a syntax that can be read by the VCS engine. Specifically, hacf translates the contents of the main configuration file, main.cf, into commands for the VCS server. You can use hacf to verify (check syntax) of main.cf and the type definition file, types.cf. VCS does not execute if hacf detects errors in the configuration. No error message and a return value of zero indicates that the syntax is legal.



## Verifying a Configuration

To verify a configuration, type:

```
# hacf -verify config_directory
```

The variable *config\_directory* refers to directories containing a main.cf file and any .cf files included in main.cf.

## Loading a Configuration

The hacf utility automatically verifies the configuration before loading it into VCS. The configuration is not loaded under the following conditions:

- ◆ If main.cf or include files are missing.
- ◆ If syntax errors appear in the .cf files.
- ◆ If the configuration file is marked “stale.” A .stale file is created in the configuration directory when you indicate that you intend to change a running configuration. See [“Setting the Configuration to Read/Write”](#) on page 89 for details.

## Dumping a Running Configuration

A configuration is dumped (written to disk) when you indicate that you have finished changing it. The configuration is also dumped on a system when the system joins the VCS cluster. When VCS dumps a running configuration, it is always pretty-printed. VCS removes the .stale file following a successful dump.

## Multiple Versions of .cf Files

When hacf creates a .cf file, it does *not* overwrite existing .cf files. A copy of the file remains in the directory, and its name includes a suffix of the date and time it was created, such as main.cf.03Dec2001.175904. In addition, the previous version of any .cf file is saved with the suffix .previous; for example, main.cf.previous.

## What is a Service Group Dependency?

A service group dependency provides a mechanism by which two service groups can be linked by a dependency rule, similar to the way resources are linked. In a service group dependency:

- ◆ A service group that depends on other service groups is a *parent group*.
- ◆ A service group on which the other service groups depend is a *child group*.
- ◆ A service group can function as both parent and child.

Parent and child service groups are linked by a *rule*. This link defines the behavior of the groups when one of them faults. A link can be configured according to the following criteria:

- ◆ The category of the dependency, such as online or offline (described in “[Categories of Service Group Dependencies](#)” on page 45).
- ◆ The location of the dependency, such as local, global, or remote (described in “[Location of Dependency](#)” on page 47).
- ◆ The type of dependency, such as soft or firm (described in “[Type of Dependency](#)” on page 49).

Based on the type of link, VCS brings the parent/child service group online or takes it offline when one of the linked service groups faults. The link also controls where VCS brings a group online following events such as a resource fault, automatic group start, system shutdown, etc.



## Why Configure a Service Group Dependency?

While defining a cluster configuration, typically a service group and an application have a one-to-one relationship. For example, a service group hosts an application, or an application is contained within a service group. In a distributed computing environment there may be multiple applications running within a cluster, and one application may depend on another. For example, a database server may have several database applications depending on its services. In such situations, it is imperative that a dependency rule be specified for how groups are brought online and taken offline.

In our example, we can define a rule that requires a database server (a child group) to be online before any or all database applications (parent group) can be brought online. We can also define a rule that requires database applications to fail over when the database server faults. For example, database applications cannot be brought online until the database server is online. If the database server faults, the database applications cannot continue to provide services.

---

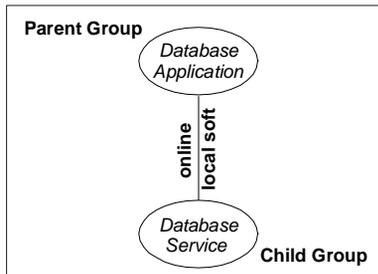
**Note** Configuring service group dependencies adds complexity to your configuration. We strongly recommend evaluating various scenarios before implementing group dependencies in your environment. In general, an application and its resources should be contained within a single service group. Group dependency helps leverage failover scenarios when multiple applications are configured in a cluster.

---

## Categories of Service Group Dependencies

### Online Group Dependency

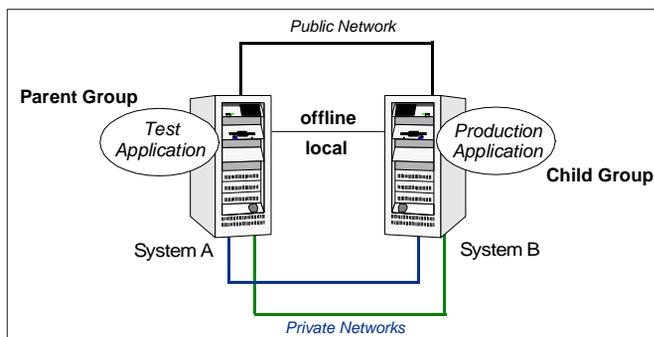
In an *online group dependency*, the parent group must wait for the child group to be brought online before it can start. For example, to configure a database application and a database service as two separate groups, you would specify the database application as the parent, and the database service as the child. The following illustration shows an *online local soft dependency* (described in “[Soft Dependency](#)” on page 49).



Online Local Soft Dependency Between a Parent Group and a Child Group

### Offline Group Dependency

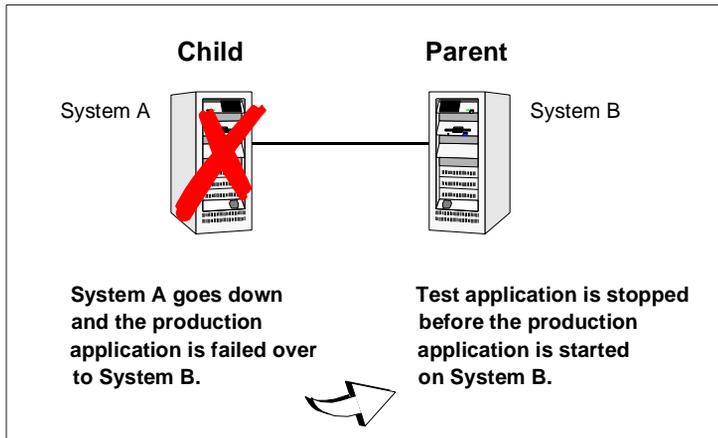
In an *offline group dependency*, the parent group can be started only if the child group is offline on the system, and vice versa. This prevents conflicting applications from running on the same system. For example, to configure a production application on one system and a test application on another, the test application must be the parent, and the production application must be the child. The following illustration shows an *offline local dependency*.



Offline Local Dependency Between a Parent Group and a Child Group



As illustrated below, System A failed while running the production application, causing the application to fail over to System B. Before VCS can restart the production application on System B, it must first stop the system's test application.



## Location of Dependency

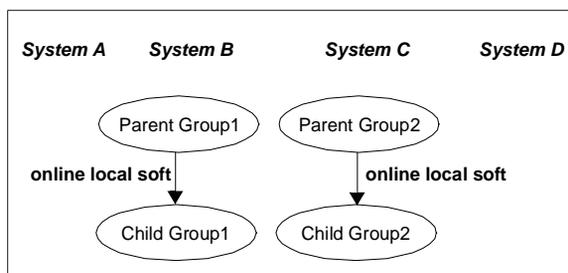
The location of the dependency determines the relative location of parent and child groups.

**Note** In the following examples, parent and child groups can be failover or parallel, as described in “Types of Service Groups” on page 15.

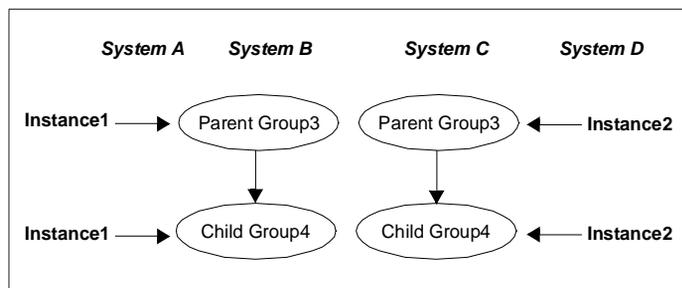
### Local Dependency

In a *local* dependency an instance of the parent group depends on an instance of the child group being online or offline on the same system, depending on the category of group dependency.

In the following figure, the failover Parent Group1 depends on failover Child Group1 online local soft. Failover Parent Group2 depends on failover Child Group2 online local soft. Failure of Child Group1 affects Parent Group1, and failure of Child Group2 affects Parent Group2.

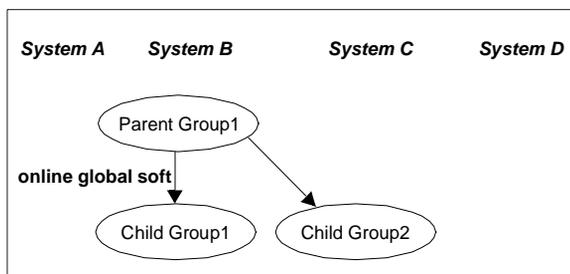


The next example of a local dependency shows that parallel Parent Group3 depends on parallel Child Group4, and Instance1 of Parent Group3 depends on an Instance1 of Child Group4.



## Global Dependency

In a *global* dependency an instance of the parent group depends on one or more instances of the child group being online on any system. In the following figure, failover Parent Group1 depends on parallel child groups online global soft. Child Group1 and Group2 are instances of the child group. In this dependency, failure of either Child Group1 or Group2 affects Parent Group1.



## Remote Dependency

In a *remote* dependency an instance of parent group depends on one or more instances of the child group being online on any system other than the system on which the parent is online.

---

## Type of Dependency

The type of dependency defines the rigidity of the link between parent and child groups. There are two dependency types: *soft* and *firm*.

### Soft Dependency

Soft dependency means VCS imposes minimal constraints while onlining parent/child groups. The only constraint is that child group *must be* online prior to the parent group being brought online. For local dependency, this implies that an instance of the child group be online on a system before an instance of the parent group can be brought online. For global dependency, an instance of the child group must be online anywhere before an instance of the parent group can be brought online. For remote dependency, an instance of the child group must be online on any other system before an instance of the parent group can be brought online.

Soft dependency provides the following enhanced flexibility:

- ◆ VCS does not immediately take the parent offline if the child group faults.
- ◆ When both groups are online, the child group can be taken offline while the parent is online and vice versa (the parent group can be taken offline while the child is online).
- ◆ The parent remains online if the child group faults and cannot fail over.
- ◆ To link a parent and child group with soft dependency, the child group is not required to be online if the parent is online. However, if the child group is also online, the parent and child may not be linked in such a way that their online states conflict with the type of link between parent and child.



The location of the link (local, global, or remote) designates whether or not a parent group will fail over after a fault and failover of the child group.

For example:

- ◆ A link configured as *online local soft* designates that when the child group faults and fails over, the parent group fails over to the same system on which the child group was brought online.
- ◆ A link configured as *online global soft* designates that when the child group faults and fails over, the parent group never fails over (remains online where it is).
- ◆ A link configured as *online remote soft* designates that when a child group faults and child group selects as the target system the system on which the parent is online, the parent group is taken offline. When a child group fails over, the parent group is brought online on another system within the cluster.

For more information regarding dependency types, see “[Overview of Service Group Dependency Configurations](#)” on page 52.

## Firm Dependency

Firm dependency means VCS imposes maximum constraints when onlining parent/child groups. Specifically:

- ◆ The child group must be online before the parent group is brought online, and, as in a soft dependency, the type of group dependency link determines if the child group can be online on the same or any other system where parent is online.
- ◆ The parent group must be taken offline when the child group faults. When the child is brought online on another system, the parent group is brought online (in accordance with the type of group dependency linking the parent and the child) on any system other than the system on which it was taken offline. For example:
  - A link configured as *online local firm* designates that the parent group is taken offline when the child group faults. When the child group fails over to another system, the parent group is brought online on the same system.
  - A link configured as *online global firm* designates that the parent group is taken offline on a system, such as System A, when the child group faults. When the child group fails over to another system, the parent group is restarted on System A.
  - A link configured as *online remote firm* designates that the parent group is taken offline when the child group faults. When the child group fails over to another system, such as System A, the parent group is migrated to a system other than System A.

In addition to the constraints imposed by soft dependency, firm dependency also includes the following constraints:

- ◆ If the child group faults, VCS takes the parent group offline as well. When the child group is brought online, the parent is brought online.
- ◆ When both groups are online, the child group cannot be taken offline while the parent group is online. However, the parent group can be taken offline while the child is online.
- ◆ If the child group faults, the parent is taken offline. If the child cannot fail over, the parent remains offline.
- ◆ To link a parent and child group with firm dependency, the parent group must be offline or the parent and child group must be online in such a way that their online states do not conflict with the type of link between parent and child.

Both soft and firm dependencies allow that if the parent group faults, the child group doesn't. The parent group may or may not fail over, depending on the link constraints (such as *online local* versus *online global*). For example, if a failover parent and failover child are linked as an online local soft/firm dependency, if the parent faults, it cannot fail over to another system. However, if parent and child were linked with an online global soft/firm or online remote soft/firm dependency, if the parent faults the parent can fail over to another system.



## Overview of Service Group Dependency Configurations

In the following sections the term “instance” applies to parallel groups only. If a parallel group is online on three systems, an instance of the group is online on each system. For failover groups, only one instance of a group is online at any time.

---

**Note** While configuring group dependencies, if dependency type (soft/firm) is omitted, the group dependency defaults to firm.

---

The following information describes situations in which a child group faults in all service group dependencies with failover and parallel groups.

### Failover Parent/Failover Child

**online local soft** Failover parent group soft depends on failover child group being online on the same system.

Parent can be brought online on a system, for example, System A, only if the child is online on System A.

- ✓ If the child faults, the parent is not taken offline. After the child successfully fails over to another system, for example, System B, VCS migrates the parent to System B. If the child cannot fail over, the parent remains online on System A.
- ✓ If parent faults on System A, child remains online on System A. Parent cannot fail over anywhere.

**online local firm** Failover parent group firm depends on failover child group being online on the same system.

Parent can be brought online on a system, for example, System A, only if the child is online on System A.

- ✓ If the child faults, the parent is taken offline on System A. When a child successfully fails over to another system, for example System B, VCS migrates the parent to System B. If child cannot fail over, parent remains offline.
- ✓ If parent faults on System A, child remains online on System A. Parent cannot fail over anywhere.

**online global soft** Failover parent group soft depends on failover child group being online anywhere in the cluster. Parent can be brought online as long as a child group is running somewhere in the cluster.

- ✓ If the child faults, the parent remains online when the child faults and fails over. The parent also remains online when the child faults and cannot fail over.
- ✓ If parent faults on System A, child remains online on System A. Parent fails over to next-available system. If no system is available, the parent remains offline.

**online global firm (default)** Failover parent group firm depends on failover child group being online anywhere in the cluster.

Parent can be brought online as long as a child group is running somewhere in the cluster. For example, the parent group is online on System A, and the child group is online on System B.

- ✓ If the child faults on System B, the parent group on System A is taken offline. When the child successfully fails over to another system, for example, System C, VCS restarts parent group on System A. If child group cannot fail over, parent group remains offline.
- ✓ If parent faults on System A, child remains online on System A. Parent fails over to next-available system. If no system is available, the parent remains offline.

**online remote soft** Failover parent group soft depends on failover child group being online on any other system in the cluster.

Parent can be brought online on any system other than the system on which the child is online. For example if child group is online on System B, the parent group can be online on System A.

- ✓ If the child faults on System B, the parent remains online on System A unless VCS selects System A as the target system on which to bring the child group online. In that case, the parent is taken offline. After the child successfully fails over to System A, VCS brings the parent online on another system, for example System B. If the child faults on System A, the parent remains online on System B unless VCS selects System B as the target system.



**online remote firm (default)** Failover parent group firm depends on failover child group being online on any other system in the cluster.

Parent can be brought online on any system other than the system on which the child is online. For example if child group is online on System A, the parent group can be online on System B.

- ✓ If the child faults on System A, the parent is taken offline on System B. After the child successfully fails over to System B, VCS brings the parent online on another system, for example, System C. However, if child fails over to System C, parent is restarted on System B.
- ✓ If the parent faults on System A, the child remains online on System B. The parent on System A fails over to a system other than A or B. If no system is available, the parent remains offline.

**offline local** Failover parent group depends on failover child group being offline on the same system and vice versa.

Parent can be brought online on any system as long as the child is not online on the system, and vice versa. For example, if child group is online on System B, the parent can be brought online on System A.

- ✓ If the child faults on System B, and if VCS selects System A as the target on which to bring the child online, the parent on System A is taken offline and the child is brought online. However, if child selects System C as the target, parent remains online on System A.
- ✓ If parent faults, child remains online. If there is no other system to which parent can fail over, parent remains offline.

## Failover Parent/Parallel Child

**online local soft** Failover parent group soft depends on an instance of the child group being online on the same system.

Failover group can be brought online on any system, for example System A, only if an instance of the child group is online on System A.

- ✓ If an instance of the child group on System A faults, the parent cannot migrate until the child has successfully failed over. After the child fails over to another system, for example, System B, the parent migrates to System B. If the instance of child cannot fail over, the parent may continue to run on System A.

Consider a configuration in which multiple instances of the child group are online on Systems A and B and the parent group is online on System A.

- ✓ If the child faults, the parent group fails over to System B.
- ✓ If the parent faults, it fails over to System B. The child on System A remains online. The parent group now depends on the instance of the child group on System B.

**online local firm (default)** Failover parent group firm depends on an instance of the child group being online on the same system.

Failover group can be brought online on any system, for example, System A, only if an instance of the child group is online on System A.

- ✓ If the instance of the child group on System A faults, the parent is taken offline. After the child has successfully failed over to another system, for example System B, the parent then fails over to System B.

Consider a configuration in which multiple instances of the child group are online on Systems A and B and the parent group is online on System A.

- ✓ If the parent faults, it fails over to System B. The child on System A remains online. The parent group now depends on the instance of the child group on System B.

**online global soft** Failover parent group soft depends on all instances of the child group being online anywhere in the cluster.

Failover group can be brought online anywhere as long as all instances of the child group are online somewhere in the cluster.

- ✓ If one or all instances of the child group fault, the parent remains online.

Consider that multiple instances of the child group are online on Systems A and B, and the parent group is online on System A.

- ✓ If parent faults, it fails over to System B. Both instances of the child group remain online, and the parent group maintains its dependency on the instances.



**online global firm (default)** Failover parent group firm depends on all instances of the child group being online anywhere in the cluster.

Failover group can be brought online anywhere as long as all instances of the child group are online somewhere in the cluster. For example, if two instances of the child are online on Systems A and B, and the parent is online on System A, if an instance of the child group faults, the parent is taken offline on System A. After the child has successfully failed over to System C, VCS restarts the parent group on System A. If the instance of the child group cannot fail over, the parent may not be brought online.

Consider that multiple instances of the child group are online on Systems A and B, and the parent group is online on System A.

- ✓ If parent faults, it fails over to System B. Both instances of the child group remain online, and the parent group maintains its dependency on the instances.

**online remote soft** Failover parent group soft depends on all instances of the child group being online on another system in the cluster.

Parent can be brought online on any system other than the system on which the child is online. For example if child group is online on Systems A and C, the parent group can be online on System B.

- ✓ If the child faults on System A, the parent remains online on System B unless VCS selects System B as the target system. After the child successfully fails over to System B, VCS brings the parent online on another system, for example, System D.
- ✓ If parent group faults on System B, both instances of the child group remain online. The parent group fails over to System D and maintains its dependency on both instances of the child group.

**online remote firm (default)** Failover parent group firm depends on all instances of the child group being online on another system in the cluster.

Failover group can be brought online anywhere as long as all instances of the child group are online on another system. For example, if a child group is online on System A and System C, the parent group can be online on System B. When the child group on System A faults, the parent is taken offline. After the child has successfully failed over to System B, VCS brings the parent online on another system, for example, System D. If the child group fails over to System D, the parent group is restarted on System B.

---

**Note** System D is selected as an example only. The parent may be restarted on Systems A, B, or D, depending on the value of the FailOverPolicy attribute for the parent group and the system on which the child group is online.

---

- ✓ If parent group faults on System B, both instances of the child group remain online. The parent group fails over to System D and maintains its dependency on both instances of the child group.

**offline local** Failover parent group depends on no instances of the child group being online on the same system, and vice versa.

Failover group can be brought online anywhere as long as any instances of the child group are not online on that system, and vice versa. For example, if the child group is online on Systems B and C, the parent group can be brought online on System A. If the child group faults on System C, and if VCS selects System A as the target on which to bring the child group online, the parent group on System A is taken offline and the child is brought online. However, if the child group selects System D as the target, the parent group remains online on System A.

- ✓ If the parent group faults, the child group remains online. If there is no other system to which the parent can fail over, the parent remains offline.

## Parallel Parent/Failover Child

**online global soft** All instances of parent group soft depend on failover group.

All instances of the parent group can be online anywhere as long as the child is online somewhere in the cluster. An instance of the parent group does not fault if an instance of the child group faults.

**online global firm (default)** All instances of parent group firm depend on failover group.

All instances of the parent group can be online anywhere as long as the child is online on another system. For example, the child group is online on System A, the parent group is online on Systems A and B.

- ✓ If the child faults, all instances of the parent group are taken offline on Systems A and B. After the child has successfully failed over to System B, VCS restarts all instances of the parent group on Systems A and B.
- ✓ If an instance of the parent group on System A faults, the child group remains online, and the parent group fails over to System C.

**online remote soft** All instances of parent group soft depend on failover group on any other system.

An instance of the parent group can be online anywhere as long as the child is online on another system. For example, the child group is online on System A, the parent group can be online on System B and System C.

- ✓ If the child group faults and VCS selects System B as the target on which to bring the child online, the instance of the parent group running on System B is taken offline. After the child has successfully failed over to System B, VCS brings online the failed parent instance to another system, for example, System D.



However, if the child group failed over to System D, the parent remains online. (If parent group on System B faults, it fails over to System D. The child group remains online on System A.)

**online remote firm (default)** All instances of parent group firm depend on failover group on any other system.

An instance of the parent group can be online anywhere as long as the child is online on another system. For example, if the child group is online on System A, the parent group can be online on System B and System C.

- ✓ If the child faults, all instances of the parent group are taken offline on System B and System C. After the child has successfully failed over to System B, VCS restarts the parent instance on System C and fails over a parent instance on System A.

However, if the child group fails over to System D, VCS restarts the parent instances on Systems B and C. (If parent group on System B faults, it fails over to System D. The child group remains online on System A.)

**offline local** All instances of the parent group depend on the child group being offline on that system and vice versa.

An instance of the parent group can be brought online anywhere as long as the child is not online on the system, and vice versa. For example, if the child group is online on System A, the parent group can be online on System B and System C.

- ✓ If the child faults on System A, and if VCS selects System B as the target on which to bring the child online, the parent on System B is taken offline first. However, if the child fails over to System D, the parent group remains online on Systems B and C.
- ✓ If the parent group faults on System B, the child group remains online and the parent group fails over to System D.

## Parallel Parent/Parallel Child

**online local soft** An instance of the parent group soft depends on an instance of the child group on the same system.

An instance of a parent group can be brought online on a system, for example, System A, only if an instance of a child group is online on System A. For example, two instances of the parent are online on System A and System B, and each instance depends on an instance of the child being online on the same system.

- ✓ If the instance of the child group on System A faults, the child group fails over to System C. After the child fails over to another system, the instance of the parent group on System A also fails over to System C. If the child cannot fail over, the parent remains online. Other instances of the parent group are unaffected.
- ✓ If an instance of the parent group on System B faults, it can fail over to System C only if an instance of the child group is running on System C and no instance of the parent group is running on System C.

**online local firm** An instance of the parent group firm depends on an instance of the child group on the same system.

An instance of a parent group can be brought online on a system, for example, System A, only if an instance of a child group is online on System A. For example, two instances of the parent are online on System A and System B, and each instance depends on an instance of the child being online on the same system.

- ✓ If an instance of the child group on System A faults, the instance of the parent group on System A is taken offline. After the child fails over to another system, for example, System C, VCS brings an instance of the parent group online on System C. Other instances of the parent group are unaffected.
- ✓ If an instance of the parent group on System B faults, it can fail over to System C only if an instance of the child group is running on System C and no instance of the parent group is running on System C.

**offline local** An instance of a parent group depends on an instance of a child group being offline on the same system and vice versa.

An instance on a system of a parent group can be brought online provided that an instance of the child is not online on the same system and vice versa. For example, if the child group is online on System C and System D, the parent can be online on System A and System B.

- ✓ If the child on System C faults and VCS selects System A as the target on which to bring the child group online, the instance of the parent on System A is taken offline first.
- ✓ When an instance of a child group or parent group faults, it has no effect on the other running instances.



## Configuring Service Group Dependencies

To configure a service group dependency, place the `requires` clause in the service group declaration within the VCS configuration file, before the resource dependency specifications, and after the resource declarations. For example:

- ◆ To configure `groupx` and `groupy` as an online local firm dependency:

```
group groupx (...group definition...)...resource declarations...
requires group groupy online local firm...resource dependencies...
```

- ◆ To configure `groupx` and `groupy` as an online global soft dependency:

```
group groupx (...group definition...)...resource declarations...
requires group groupy online global soft...resource dependencies...
```

- ◆ To configure `groupx` and `groupy` as an online remote soft dependency:

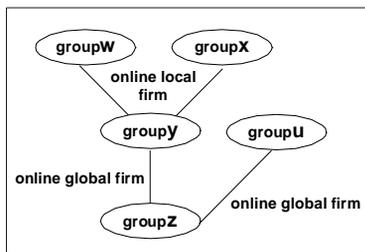
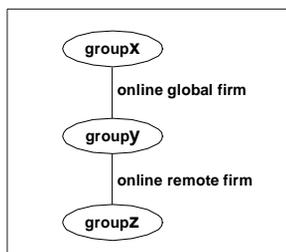
```
group groupx (...group definition...)...resource declarations...
requires group groupy online remote soft...resource dependencies...
```

- ◆ To configure `groupx` and `groupy` as an offline local dependency:

```
group groupx (...group definition...)...resource declarations...
requires group groupy offline local...resource dependencies...
```

## Dependency Limitations

Each parent group can link with only one child group; however, a child group can have multiple parents. A service group dependency tree can have three levels, maximum. For example, in the illustration below left, `groupX` requires `groupY` online global firm, and `groupY` requires `groupZ` online remote firm. In the illustration below right, `groupW` requires `groupY` online local firm, `groupX` requires `groupY` online local firm, `groupY` requires `groupZ` online global firm, and `groupU` requires `groupZ` online global firm.



## Automatic Actions for Service Group Dependencies

### Automatic Online

If a service group is configured to start automatically on a system, it is brought online only if the group's dependency requirements are met. This implies that in an online local dependency, parent groups are brought online only after all child groups are brought online.

### AutoRestart

If a persistent resource on a service group (group1 in this example) faults, the service group is automatically failed over to another system in the cluster under the following conditions:

- ◆ The AutoFailover attribute is set.
- ◆ There is another system in the cluster to which group1 can fail over.

If neither of the above conditions is met (the AutoFailover attribute is not set or other systems in the cluster are unavailable), group1 remains offline and faulted, even after the faulted resource becomes online.

Setting the AutoRestart attribute enables a service group to be brought back online without manual intervention. In the above example, setting the AutoRestart attribute for the group1 would enable VCS to bring the group back online, after the resource came online on the system where the resource faulted.

Or, if group1 could not fail over to another system because none was available, setting the AutoRestart attribute would enable VCS to bring the group back online on the first available system after the group's faulted resource came online.

For example, NIC is a persistent resource. In some cases, when a system boots and VCS starts, VCS probes all resources on the system. It is possible that when VCS probes the NIC resource, the resource may not yet be online because the networking is not up and fully operational. When this occurs, VCS will mark the NIC resource as faulted, and will not bring the service group online. However, when the NIC resource becomes online and if AutoRestart is enabled, the service group is brought online.



## Automatic Failover

A failover occurs when a service group faults and is migrated to another system. It can also occur when a system crashes and the groups running on that system migrate to other systems in the cluster. For service groups with dependencies, the following actions occur during failover:

- ✓ A target system is selected on which the service group's dependency requirements are met.
- ✓ If a target system exists, but there is a dependency violation between the service group and a parent group, the parent group is migrated to another system to accommodate the service group that is moving to the target system. In conflicts between a child group and a parent group, the child group takes priority.
- ✓ If the service group has a parent with an online local firm dependency, when the child group faults, the parent group is taken offline. When the child successfully fails over to another system, the parent is brought online.
- ✓ If the service group has a parent with an online local soft dependency, when the child group faults, the parent group remains online. When the child successfully fails over to another system, the parent migrates to that system.
- ✓ For soft dependencies, when child group faults and cannot fail over the parent group remains online.
- ✓ For firm dependencies, when child group faults and cannot fail over the parent group remains offline and no further attempt is made to bring it online.

## Manual Operations for Service Group Dependencies

As described below, the `hagrp -online`, `-offline`, and `-switch` commands enable you to manually bring a service group online, take it offline, or fail it over.

### Manual Online

Basic rules governing how to manually bring a service group online also apply to service groups with dependencies. Additionally, the following rules apply for service groups configured with dependencies. For example:

- ◆ For online dependencies, a parent group cannot be brought online manually if the child is not online.
- ◆ For online local dependencies, a parent group cannot be brought online manually on any system other than the system on which the child is online.
- ◆ For online remote dependencies, a parent group cannot be brought online manually on the system on which the child is online.
- ◆ For offline local dependencies, a parent group cannot be brought online manually on the system on which the child is online.

Typically, bringing a child group online manually is never rejected, except under the following circumstances:

- ◆ For online local dependencies, if parent is online, a child group online is rejected for any system other than the system where parent is online.
- ◆ For online remote dependencies, if parent is online, a child group online is rejected for the system where parent is online.
- ◆ For offline local dependencies, if parent is online, a child group online is rejected for the system where parent is online.

The following examples describe situations where bringing a parallel child group online is accepted:

- ◆ For a parallel child group linked online local with failover/parallel parent, multiple instances of child group online are acceptable.
- ◆ For a parallel child group linked online remote with failover parent, multiple instances of child group online are acceptable, as long as child group does not go online on the system where parent is online.
- ◆ For a parallel child group linked offline local with failover/parallel parent, multiple instances of child group online are acceptable, as long as child group does not go online on the system where parent is online.



## Manual Offline

Basic rules governing how to manually take a service group offline also apply to service groups with dependencies. Additionally, VCS rejects manual offlining if the procedure violates existing group dependencies. Typically, firm dependencies are more restrictive to offlining a child group while parent group is online. Rules for manual offlining include:

- ◆ Parent group offline is never rejected.
- ◆ For all firm dependencies, if parent group is online, child group offline is rejected.
- ◆ For all soft dependencies, child group can be offlined regardless of the state of parent group.

## Manual Switch

Switching a service group implies manually taking a service group offline on one system, and manually bringing it back online on another system. Basic rules governing how to manually switch a service group also apply to service group dependencies. Additionally, VCS rejects manual switch if the group does not comply with manual offline or manual online rules described above.

## Linking Service Groups (Online/Offline Dependencies)

As described previously, a configuration may require that a certain service group be running before another service group can be brought online. For example, a group containing resources of a database service must be running before the database application is brought online.

To specify this dependency, type:

```
# hagrp -link parent_group child_group gd_category  
          gd_location gd_type
```

The variable *parent\_group* is the name of a service group.

The variable *child\_group* is the name of a service group.

The variable *gd\_category* is the category of group dependency (online/offline).

The variable *gd\_location* is the boundary of *parent\_group-child\_group* link (local/global/remote).

The optional variable *gd\_type* is the type of group dependency (soft/firm).

The *parent\_group* is linked to the *child\_group* by a link that is described by a combination of *gd\_category*, *gd\_location* and *gd\_type*.

## Constraints

- ◆ Each parent group can link with only one child group; however, a child group can have multiple parents.
- ◆ A service group dependency tree can have three levels, maximum.
- ◆ You cannot link two service groups whose current states violate the relationship.
  - All link requests are accepted if all instances of parent group are offline.
  - All online local link requests are rejected if for an instance of parent group, an instance of child group is not online on the same system.
  - All online remote link requests are rejected when an instance of parent group and an instance of child group are running on the same system.
  - All offline local link requests are rejected when an instance of parent group and an instance of child group are running on the same system.
  - All link requests are rejected, if parent group is online and child group is offline.
  - All online global/online remote link requests to link two parallel groups are rejected.
  - All online local link requests to link a parallel parent group to failover child group are rejected.



## Dependency Summary Sheet and FAQs

The following matrices depict actions performed by parent and child groups according to dependency type and location, and if a failover system is targeted for the group. This section also includes a list of frequently asked questions (FAQs) regarding each group location.

### Online Local

Online Local		Failover System for Group	No Failover System for Group
Parent Fails	Firm	<ul style="list-style-type: none"> <li>- Parent faults.</li> <li>- No failover for parent.</li> <li>- Child continues to run on original system.</li> </ul>	<ul style="list-style-type: none"> <li>- Parent faults.</li> <li>- No failover for parent.</li> <li>- Child continues to run on original system.</li> </ul>
	Soft	<ul style="list-style-type: none"> <li>- Parent faults.</li> <li>- No failover for parent.</li> <li>- Child continues to run on original system.</li> </ul>	<ul style="list-style-type: none"> <li>- Parent faults.</li> <li>- No failover for parent.</li> <li>- Child continues to run on original system.</li> </ul>
Child Fails	Firm	<ul style="list-style-type: none"> <li>- Child faults.</li> <li>- Parent taken offline.</li> <li>- Child fails over and starts.</li> <li>- Parent starts on system with child.</li> </ul>	<ul style="list-style-type: none"> <li>- Child faults.</li> <li>- Parent taken offline.</li> <li>- Both groups die.</li> </ul>
	Soft	<ul style="list-style-type: none"> <li>- Child faults.</li> <li>- Child fails over to available system.</li> <li>- Parent fails over to same system as child.</li> </ul>	<ul style="list-style-type: none"> <li>- Child faults.</li> <li>- No failover for parent.</li> <li>- Parent continues to run on original system.</li> </ul>



## FAQ for Online Local Dependency

Can parent group be brought online when child group is offline? Firm=No Soft=No.

Can child group be taken offline when parent group is online? Firm=No Soft=Yes.

Can parent group be switched while child group is running? Firm=No Soft=No.

Can child group be switched while the parent group is running? Firm=No Soft=Yes.  
(Parent then switches after child.)

## Online Global

Online Global		Failover System for Group	No Failover System for Group
Parent Fails	Firm	<ul style="list-style-type: none"> <li>- Parent faults.</li> <li>- Parent fails over.</li> <li>- Child continues to run on original system.</li> </ul>	<ul style="list-style-type: none"> <li>- Parent faults.</li> <li>- Parent dies.</li> </ul>
	Soft	<ul style="list-style-type: none"> <li>- Parent faults.</li> <li>- Parent fails over.</li> <li>- Child continues to run.</li> </ul>	<ul style="list-style-type: none"> <li>- Parent faults.</li> <li>- Parent dies.</li> </ul>
Child Fails	Firm	<ul style="list-style-type: none"> <li>- Child faults.</li> <li>- Parent taken offline.</li> <li>- Child fails over.</li> <li>- Parent restarts on system on which it was online.</li> </ul>	<ul style="list-style-type: none"> <li>- Child faults.</li> <li>- Parent taken offline.</li> <li>- Both groups die.</li> </ul>
	Soft	<ul style="list-style-type: none"> <li>- Child faults.</li> <li>- Child fails over.</li> <li>- Parent continues to run on original system.</li> </ul>	<ul style="list-style-type: none"> <li>- Child faults.</li> <li>- Child dies.</li> </ul>

## FAQ for Online Global Dependency

Can parent group be brought online when child group is offline? Firm=No Soft=No.

Can child group be taken offline when parent group is online? Firm=No Soft=Yes.

Can parent group be switched while child group is running? Firm=Yes Soft=Yes.

Can child group be switched while the parent group is running? Firm=No Soft=Yes.



## Online Remote

Online Remote		Failover System for Group	No Failover System for Group
Parent Fails	Firm	<ul style="list-style-type: none"> <li>- Parent faults.</li> <li>- Parent fails over to system without child. If the only system available is where child is running, parent is not brought online.</li> </ul>	<ul style="list-style-type: none"> <li>- Parent faults.</li> <li>- Parent dies.</li> <li>- Child continues running.</li> </ul>
	Soft	<ul style="list-style-type: none"> <li>- Parent faults.</li> <li>- Parent fails over to system without child. If the only system available is where child is running, parent is not brought online.</li> </ul>	<ul style="list-style-type: none"> <li>- Parent faults.</li> <li>- Parent dies.</li> <li>- Child continues running.</li> </ul>
Child Fails	Firm	<ul style="list-style-type: none"> <li>- Child faults.</li> <li>- Parent taken offline.</li> <li>- Child fails over.</li> <li>- If child fails over to the system on which the parent was online, the parent restarts on a system different from the child. Otherwise, parent restarts on original system.</li> </ul>	<ul style="list-style-type: none"> <li>- Child faults.</li> <li>- Parent taken offline.</li> <li>- Both groups die.</li> </ul>
	Soft	<ul style="list-style-type: none"> <li>- Child faults.</li> <li>- Child fails over. If child fails over to the system on which parent was online, the parent restarts on a system different from the child. Otherwise, parent restarts on original system.</li> </ul>	<ul style="list-style-type: none"> <li>- Child faults.</li> <li>- Child dies.</li> <li>- Parent continues running.</li> </ul>



## FAQ for Online Remote Dependency

Can parent group be brought online when child group is offline? Firm=No Soft=No.

Can child group be taken offline when parent group is online? Firm=No Soft=Yes.

Can parent group be switched while the child group is running? Firm=Yes, but not to system on which child is running. Soft=Yes, but not to system on which child is running.

Can child group be switched while the parent group is running? Firm=No Soft=Yes, but not to system on which parent is running.

## Offline Local

Offline Local	Failover System for Group	No Failover System for Group
Parent Fails	<ul style="list-style-type: none"> <li>- Parent faults.</li> <li>- Parent fails over to system without child.</li> </ul>	<ul style="list-style-type: none"> <li>- Parent faults.</li> <li>- Parent dies.</li> <li>- Child continues running.</li> </ul>
Child Fails	<ul style="list-style-type: none"> <li>- Child faults.</li> <li>- If child fails over to system on which parent is running, parent is taken offline.</li> <li>- If parent is taken offline, it starts on another system, if available.</li> </ul>	<ul style="list-style-type: none"> <li>- Child faults.</li> <li>- Parent continues running.</li> </ul> <p>(This happens if child group is already faulted on the system where parent was running. Child has no available systems.)</p>
	<ul style="list-style-type: none"> <li>- Child faults.</li> <li>- Child fails over. Otherwise, parent continues running.</li> </ul>	N/A

## FAQ for Offline Local Dependency

Can parent group be brought online when child group is offline? Yes.

Can child group be taken offline when parent group is online? Yes.

Can parent group be switched while the child group is running? Yes, but not to system on which child is running.

Can child group be switched while the parent group is running? Yes, but not to system on which parent is running.







## **Section III Administration—Putting VCS to Work**

---

This section describes VCS user privileges and how they are used to control access to the cluster. It also provides information on monitoring and administering VCS from the graphical-user interfaces and the command line.

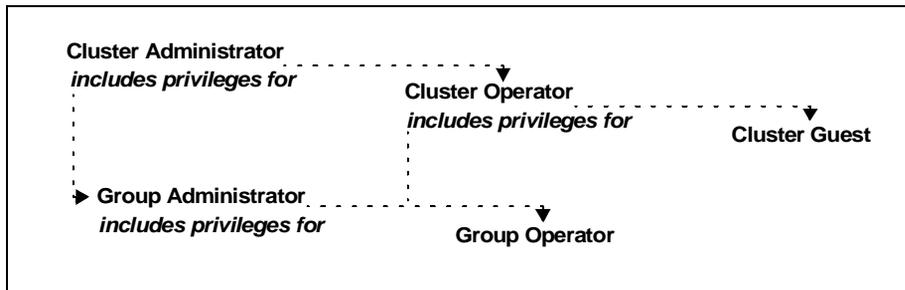
Section III includes the following chapters:

- ◆ **Chapter 5 “Controlling Access to VCS” 73**
- ◆ **Chapter 6 “Administering VCS from the Command Line” 83**
- ◆ **Chapter 7 “Administering VCS from Cluster Manager (Java Console)” 117**
- ◆ **Chapter 8 “Administering VCS from Cluster Manager (Web Console)” 203**

## User Privileges

Cluster operations are enabled or restricted depending on the permissions with which you log on to VCS. There are various privilege levels, or “categories,” for users administering VCS. Each category is assigned specific privileges, and some categories overlap; for example, Cluster Administrator includes privileges for Group Administrator, which includes privileges for Group Operator. The category Cluster Guest has the fewest privileges, Cluster Administrator the most. For instructions on how to add a user and assign privileges, see [page 150](#).

The following illustration shows the categories of user privileges and how they overlap with one another.



The five user categories are listed below, along with a summary of their associated privileges.

---

User Category	Privileges
Cluster Administrator	<p>Users in this category are assigned full privileges, including making configuration read-write, creating and deleting groups, setting group dependencies, adding and deleting systems, and adding, modifying, and deleting users. All group and resource operations are allowed. Users can also change other user's privileges.</p> <p>Additionally, users in this category can create and delete resource types and execute remote commands from Cluster Manager via Cluster Shell.</p>
Cluster Operator	<p>In this category, all cluster-, group-, and resource-level operations are allowed, including modifying the user's own password and bringing service groups online. Note that users in this category cannot create service groups or execute remote commands via Cluster Shell.</p> <p>Additionally, users in this category can be assigned Group Administrator privileges for specific service groups.</p>
Group Administrator	<p>Users in this category can perform all service group operations, such as bringing groups and resources online, taking them offline, and creating or deleting resources for the group. Additionally, users can establish resource dependencies and freeze or unfreeze persistent and non-persistent service groups. Note that users in this category cannot create or delete service groups.</p>
Group Operator	<p>Users in this category can bring service groups and resources online and take them offline. Users can also temporarily freeze or unfreeze service groups.</p>

User Category	Privileges
Cluster Guest	<p>Users in this category have read-only access, meaning they can view the configuration, but cannot change it. They can modify their own passwords only if the configuration is in read/write mode. They cannot add or update users. Additionally, users in this category can be assigned Group Administrator or Group Operator privileges for specific service groups.</p> <p><b>Note</b> By default, newly created users are assigned Cluster Guest permissions.</p>

User categories are set *implicitly*, as shown in figure on [page 73](#), but may also be set *explicitly* for specific service groups. For example, a user in category Cluster Operator can be assigned the category Group Administrator for one or more service groups. Likewise, a user in category Cluster Guest can be assigned Group Administrator and Group Operator.

Review the following sample main.cf:

```
Cluster vcs
  UserNames = { sally = Y2hJtFngctD76, tom = pJad09NWtXHlk,
    betty = kjheewoieuo, lou = T6jhjFYkie, don = gt3tgfdgttU,
    intern = EG67egdsak }
  Administrators = { tom }
  Operators = { sally }
  ...
)

Group finance_server (
  Administrators = { betty }
  Operators = { lou, don }
  ...
)

Group hr_application (
  Administrators = { sally }
  Operators = { lou, betty }
  ...
)

Group test_server (
  Administrators = { betty }
  Operators = { intern, don }
  ...
)
```



- ◆ User “tom” is Cluster Administrator.
- ◆ User “sally” is Cluster Operator and Group Administrator for service group hr\_application.
- ◆ User “betty” does not have Cluster Administrator or Cluster Operator privileges. However, she is Group Administrator for the service groups finance\_server and test\_server. She is also Group Operator for the service group hr\_application.
- ◆ User “lou” has no privileges at the cluster level. However, he is Group Operator for the service groups finance\_server and hr\_application.
- ◆ User “don” does not have Cluster Administrator or Cluster Operator privileges. However, he is Group Operator for the service groups finance\_server and test\_server.
- ◆ User “intern” does not have Cluster Administrator or Cluster Operator privileges. However he or she is Group Operator for the service group test\_server.

Category	tom	sally	betty	lou	don	intern
Cluster Administrator	✓	-	-	-	-	-
Cluster Operator	✓	✓	-	-	-	-
finance_server Admin.	✓	-	✓	-	-	-
finance_server Operator	✓	✓	✓	✓	✓	-
hr_application Admin.	✓	✓	-	-	-	-
hr_application Operator	✓	✓	✓	✓	-	-
test_server Admin.	✓	-	✓	-	-	-
test_server Operator	✓	✓	✓	-	✓	✓

## Administration Matrices

Review the matrices in the following section to determine which command options can be executed within a specific user category. (Command matrices are listed alphabetically.) Checkmarks denote the command and option can be executed. A dash indicates they cannot. See [Appendix C](#) for matrices of less frequently used commands.

In general, users with Cluster Guest privileges can execute the command options -display, -state, and -value. Users with privileges for Group Operator and Cluster Operator can execute the options -online, -offline, and -switch. Users with Group Administrator and Cluster Administrator privileges can execute the options -add, -delete, and -modify.

### haclus

haclus Options	Cluster Guest	Group Operator	Group Admin.	Cluster Operator	Cluster Admin.
-display	✓	✓	✓	✓	✓
-value	✓	✓	✓	✓	✓
-modify	-	-	-	-	✓
<b>Note</b> Only users with root privileges can execute the command <code>haclus -modify HacliUserLevel</code> .					
-help	✓	✓	✓	✓	✓



## hagr

hagr Options	Cluster Guest	Group Operator	Group Admin.	Cluster Operator	Cluster Admin.
-add	-	-	-	-	✓
-delete	-	-	-	-	✓
-link	-	-	-	-	✓
-unlink	-	-	-	-	✓
-clear	-	✓	✓	✓	✓
-online	-	✓	✓	✓	✓
-offline	-	✓	✓	✓	✓
-state	✓	✓	✓	✓	✓
-switch	-	✓	✓	✓	✓
-freeze	-	✓	✓	✓	✓
-freeze -persistent	-	-	✓	-	✓
-unfreeze	-	✓	✓	✓	✓
-unfreeze -persistent	-	-	✓	-	✓
-enable	-	-	✓	-	✓
-disable	-	-	✓	-	✓
-modify	-	-	✓	-	✓
-display	✓	✓	✓	✓	✓
-dep	✓	✓	✓	✓	✓
-resources	✓	✓	✓	✓	✓
-list	✓	✓	✓	✓	✓



hagrpd Options	Cluster Guest	Group Operator	Group Admin.	Cluster Operator	Cluster Admin.
-value	✓	✓	✓	✓	✓
-enableresources	-	-	✓	-	✓
-disableresources	-	-	✓	-	✓
-flush	-	✓	✓	✓	✓
-autoenable	-	-	✓	-	✓
-help	✓	✓	✓	✓	✓

## hares

hares Options	Cluster Guest	Group Operator	Group Admin.	Cluster Operator	Cluster Admin.
-add	-	-	✓	-	✓
-delete	-	-	✓	-	✓
-local	-	-	✓	-	✓
-global	-	-	✓	-	✓
-link	-	-	✓	-	✓
-unlink	-	-	✓	-	✓
-clear	-	✓	✓	✓	✓
-online	-	✓	✓	✓	✓
-offline	-	✓	✓	✓	✓
-offprop	-	✓	✓	✓	✓
-modify	-	-	✓	-	✓



hares Options	Cluster Guest	Group Operator	Group Admin.	Cluster Operator	Cluster Admin.
-state	✓	✓	✓	✓	✓
-display	✓	✓	✓	✓	✓
-dep	✓	✓	✓	✓	✓
-list	✓	✓	✓	✓	✓
-value	✓	✓	✓	✓	✓
-probe	-	✓	✓	✓	✓
-help	✓	✓	✓	✓	✓

## hastatus

hastatus Options	Cluster Guest	Group Operator	Group Admin.	Cluster Operator	Cluster Admin.
-sound	✓	✓	✓	✓	✓
-summary	✓	✓	✓	✓	✓
-sound -group	✓	✓	✓	✓	✓



## hasys

hasys Options	Cluster Guest	Group Operator	Group Admin.	Cluster Operator	Cluster Admin.
-add	-	-	-	-	✓
-delete	-	-	-	-	✓
-freeze	-	-	-	✓	✓
-freeze -persistent	-	-	-	-	✓
-freeze -evacuate	-	-	-	-	✓
-freeze -persistent -evacuate	-	-	-	-	✓
-unfreeze	-	-	-	✓	✓
-unfreeze -persistent	-	-	-	-	✓
-display	✓	✓	✓	✓	✓
-force	-	-	-	-	✓
-load	-	-	-	-	✓
-modify	-	-	-	-	✓
-state	✓	✓	✓	✓	✓
-list	✓	✓	✓	✓	✓
-value	✓	✓	✓	✓	✓
-nodeid	✓	✓	✓	✓	✓
-help	✓	✓	✓	✓	✓



## hauser

hauser Options	Cluster Guest	Group Operator	Group Admin.	Cluster Operator	Cluster Admin.
-add	-	-	-	-	✓
-delete	-	-	-	-	✓
-update	-	✓	✓	✓	✓
-display	✓	✓	✓	✓	✓
-help	✓	✓	✓	✓	✓

## User Privileges for CLI and Cluster Shell Commands

The following information describes two important concepts for users executing commands from the command line:

- ◆ Users logged on as root are granted privileges that exceed those of Cluster Administrator, such as the ability to start and stop a cluster.
- ◆ When non-root users execute `haxxx` commands, they are prompted for their VCS user name and password to authenticate user category and associated privileges. To disable authentication, set the attribute `AllowNativeCliUsers` to 1. This instructs VCS to authenticate the user using his or her OS user name instead. Note that users must have proper cluster- and group-level privileges to execute commands. (For details, see the description of the `AllowNativeCliUsers` attribute in the Cluster attributes section of [Appendix B](#).)



# Administering VCS from the Command Line

# 6

Most commands listed in this chapter can be entered from any system in the cluster only when VCS is running. The command to start VCS is typically invoked at system startup. Instructions on how to start VCS begin on [page 85](#).

## VCS Environment Variables

Variable	Definition	Default
VCS_CONF	Root directory for VCS configuration files.	/etc/VRTSvcs <b>Note</b> If this variable is added or modified you must reboot the system to apply the changes.
VCS_ENABLE_LDF	Designates whether or not log data files (LDFs) are generated. If set to 1, LDFs are generated. If set to 0, they are not.	1
VCS_HOME	Root directory for VCS executables.	/opt/VRTSvcs
VCS_GAB_PORT	GAB port to which VCS connects.	h
VCS_GAB_TIMEOUT	Timeout in milliseconds for HAD to send heartbeats to GAB.	15000 <b>Note</b> If the specified timeout is exceeded, GAB kills HAD, and all active service groups on system are disabled.
VCS_HAD_RESTART_TIMEOUT	User must set this variable to designate the amount of time the hashadow process waits (“sleep time”) before restarting HAD.	0



Variable	Definition	Default
VCS_LOG	Root directory for log files and temporary files.	/var/VRTSvcs <b>Note</b> If this variable is added or modified you must reboot the system to apply the changes.
VCS_PORT	Port number of the VCS engine (HAD).	14141
VCS_TEMP_DIR	Directory in which temporary information required by, or generated by, hacf is stored.	/var/VRTSvcs <b>Note</b> This directory is created in /tmp under the following conditions: <ul style="list-style-type: none"> <li>- The variable is not set.</li> <li>- The variable is set but the directory to which it is set does not exist.</li> <li>- The utility hacf cannot find the default location.</li> </ul>

## How VCS Identifies the Local System

VCS checks \$VCS\_CONF/conf/sysname. If this file does not exist, the local system is identified by its node name. To view the system's node name, type **uname -n**.

## Installing a VCS License

The utility `halic` installs a new permanent license, or updates a demo license, while `HAD` is running. You must have root privileges to use this utility. This utility must be run on each system in the cluster: it cannot install or update a license on remote nodes.

### ▼ To install a new license

```
# halic key
```

The variable `key` represents the license key to be installed on the local system.

---

**Note** The utility `halic` must be run on each system in the cluster.

---

## Starting VCS

The command to start VCS is invoked from the file `/etc/rc.d/rc2.d/S99vcs`. When VCS is started on a system, it checks the state of its local configuration file and registers with GAB for cluster membership. If the local configuration is valid, and if no other system is running VCS, it builds its state from the local configuration file and enters the `RUNNING` state.

Type the following command to start VCS:

```
# hstart [-stale|-force]
```

Note that `-stale` and `-force` are optional. The option `-stale` instructs the engine to treat the local configuration as stale even if it is valid. The option `-force` instructs the engine to treat a stale, but otherwise valid, local configuration as valid.

If all systems are in `ADMIN_WAIT`, enter the following command from any system in the cluster to force VCS to use the configuration file from the system specified by the variable `system`:

```
# hasys -force system
```

When VCS is started on a system, and when that system is the only one running, VCS retrieves the configuration from the local configuration directory `$VCS_CONF/conf/config`.

If the local configuration is valid, the VCS engine performs a `LOCAL_BUILD`, and the system transitions to the state of `RUNNING`, its normal operational state. If the local configuration is missing, invalid, or designated “stale,” the system transitions to the state of `STALE_ADMIN_WAIT`, and the VCS engine waits for manual intervention, or for VCS to be started on a system that has a valid configuration.



If VCS is started on a system when other systems are already running VCS, the engine processes exchange their operational states according to the following conventions:

- ◆ If a system running VCS is in the state of `RUNNING`, the system joining the cluster performs a `REMOTE_BUILD` from that system and transitions to the state of `RUNNING`.
- ◆ If a system running VCS is in the state of `LOCAL_BUILD`, the system joining the cluster waits for that system to transition to `RUNNING`. It then performs a `REMOTE_BUILD` from that system and transitions to the state of `RUNNING`.
- ◆ If all systems running VCS are in the state of `STALE_ADMIN_WAIT`, and if the local configuration file of the system joining the cluster is valid, the joining system performs a `LOCAL_BUILD` and transitions to `RUNNING`. The other systems then perform `REMOTE_BUILDS` from the new system and transition to `RUNNING`.
- ◆ If all systems running VCS are in the state of `STALE_ADMIN_WAIT`, and if the local configuration file of the system joining the cluster is invalid, then the joining system also transitions to `STALE_ADMIN_WAIT`.

---

**Note** See [Appendix A](#) for a complete list of VCS system states and transitions.

---

## Starting VCS on a Single Node

Type the following command to start an instance of VCS that does not require the GAB and LLT packages. Do not use this command on a multisystem cluster.

```
# hstart -onode
```

## Starting VCS as Time-Sharing Process

Type the following command to start VCS as a time-sharing process:

```
# hstart -ts
```

## Stopping VCS

The `hastop` command stops HAD and related processes. This command includes the following options:

```
hastop -local [-force | -evacuate]
hastop -sys system [-force | -evacuate]
hastop -all [-force]
```

The option `-evacuate`, when combined with `-local` or `-sys`, migrates the system's active service groups to another system in the cluster, before the system is stopped.

The option `-local` stops HAD on the system you typed the command.

The option `-sys` stops HAD on the system you specified.

The option `-all` stops HAD on all systems in the cluster.

The option `-force` allows HAD to be stopped without offlining service groups on that system.

## Stopping VCS Without `-force` Option

When VCS is stopped on a system without using the `-force` option to `hastop`, it enters the LEAVING state, and waits for all groups to go offline on the system. Use the output of the command `hasys -display system` to verify that the values of the `SysState` and the `OnGrpCnt` attributes are non-zero. VCS continues to wait for the service groups to go offline before it shuts down. See “[Troubleshooting Resources](#)” on page 320 for more information.

## Stopping VCS with Options Other Than `-force`

When VCS is stopped by options other than `-force` on a system with online service groups, the groups running on the system are taken offline and remain offline. This is indicated by VCS setting the attribute `IntentOnline` to 0. Using the option `-force` enables service groups to continue running while HAD is brought down and restarted (`IntentOnline` remains unchanged).



## Additional Considerations for Stopping VCS

- ◆ If using the command `reboot`, behavior is controlled by the `ShutdownTimeOut` parameter. After `HAD` exits, if `GAB` exits within the time designated in the `ShutdownTimeout` attribute, the remaining systems recognize this as a reboot and fail over service groups from the departed system. For large systems, consider increasing the value in the `ShutdownTimeout` attribute.
- ◆ Stopping VCS on a system autodisables each service group that include the system in their `SystemList` attribute. (This does not apply to systems that are powered off.)
- ◆ If you use the `-evacuate` option, evacuation occurs before VCS is brought down.



# Adding, Modifying, and Deleting Users

## Guidelines

- ✓ The VCS configuration must be in read/write mode.
- ✓ You can add, modify, and delete users on any system in the cluster.

---

**Note** You must add users to the VCS configuration to monitor and administer VCS from the graphical user interface “Cluster Manager.”

---

## Setting the Configuration to Read/Write

The commands to add, modify, and delete a user change the attributes stored in the .cf files. Therefore, these commands can be executed only as root, and only if the VCS configuration is in read/write mode.

To set the mode to read/write, type the following command from any system in the cluster:

```
# haconf -makerw
```

In addition to setting the configuration to read/write, this command designates the configuration stale by creating the default file \$VCS\_CONF/conf/config/.stale on all systems running VCS.

## Setting the Configuration to Read-Only

When you have completed adding, modifying, and deleting users, reset the configuration to read-only:

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

In addition to setting the configuration to read-only, this command writes, or “dumps,” the configuration to disk and removes the configuration’s designation of stale.



## Adding a User with Cluster Guest Access

1. Set the configuration to read/write mode:

```
# haconf -makerw
```

2. Add the user:

```
# hauser -add user
```

3. Enter a password when prompted.

4. Reset the configuration to read-only:

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

---

**Note** Users in the category Cluster Guest cannot add users.

---

## Adding a User with Cluster Administrator Access

1. Create a user with Cluster Guest access (see “[Adding a User with Cluster Guest Access](#)”).

2. Add the user as Cluster Administrator:

```
# haclus -modify Administrators -add user
```

## Adding a User with Cluster Operator Access

1. Create a user with Cluster Guest access (see “[Adding a User with Cluster Guest Access](#)”).

2. Add the user as Cluster Operator:

```
# haclus -modify Operators -add user
```

## Adding a User with Group Administrator Access

1. Create a user with Cluster Guest access (see “[Adding a User with Cluster Guest Access](#)”).
2. Add the user as Group Administrator:

```
# hagrps -modify service_group Administrators -add user
```

## Adding a User with Group Operator Access

1. Create a user with Guest access (see “[Adding a User with Cluster Guest Access](#)”).
2. Add the user as Group Operator:

```
# hagrps -modify service_group Operators -add user
```

## Modifying a User

1. Set the configuration to read/write mode:
2. Modify the user:
3. Enter a new password when prompted.
4. Reset the configuration to read-only:

```
# haconf -makerw
```

```
# hauser -update user
```

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

---

**Note** Users in the category Cluster Guest cannot modify users.

---



## Deleting a User

1. Set the configuration to read/write mode:

```
# haconf -makerw
```

2. Delete the user from the list of registered users:

```
# hauser -delete user
```

3. Reset the configuration to read-only:

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

## Displaying a User

Type the following command to display a list of users:

```
# hauser -display
```

## Querying VCS

VCS enables you query various cluster objects, including resources, service groups, systems, resource types, agents, and clusters. You may enter query commands from any system in the cluster. Commands to display information on the VCS configuration or system states can be executed by all users: you do not need root privileges.

## Querying Service Groups

- ▼ To display the state of a service group on a system

```
# hagr -state [service_group] -sys [system]
```

- ▼ For a list of a service group's resources

```
# hagr -resources [service_group]
```

- ▼ For a list of a service group's dependencies

```
# hagr -dep [service_group]
```



---

▼ **To display a service group on a system**

```
# hagrps -display [service_group] -sys [system]
```

If *service\_group* is not specified, information regarding all service groups is displayed.

▼ **To display attributes of a system**

```
# hagrps -display [service_group] [-attribute attribute]
[-sys system]
```

---

**Note** System names are case-sensitive.

---

## Querying Resources

▼ **For a list of a resource's dependencies**

```
# hares -dep [resource]
```

▼ **For information on a resource**

```
# hares -display [resource]
```

If *resource* is not specified, information regarding all resources is displayed.

▼ **To display details of an attribute**

```
# hares -display -attribute [attribute]
```

▼ **To confirm an attribute's values are the same on all systems**

```
# hares -global resource attribute value ... | key... |
{key value}...
```

▼ **To display resources of a service group**

```
# hares -display -group [service_group]
```

▼ **To display resources of a resource type**

```
# hares -display -type [resource_type]
```

▼ **To display attributes of a system**

```
# hares -display -sys [system]
```



## Querying Resource Types

▼ For a list of resource types

```
# hatype -list
```

▼ For a list of all resources of a particular type

```
# hatype -resources resource_type
```

▼ For information about a resource type

```
# hatype -display [resource_type]
```

If *resource\_type* is not specified, information regarding all types is displayed.

## Querying Resource Type Agents

▼ For an agent's run-time status

```
# haagent -display [agent]
```

If *agent* is not specified, information regarding all agents is displayed.

Run-Time Status	Definition
Faults	Indicates the number of agent faults and the time the faults began.
Messages	Displays various messages regarding agent status.
Running	Indicates the agent is operating.
Started	Indicates the file is executed by the VCS engine (HAD).

## Querying Systems

- ▼ For a list of systems in the cluster

```
# hasys -list
```

- ▼ For information about each system

```
# hasys -display [system]
```

## Querying Clusters

- ▼ For the value of a specific cluster attribute

```
# haclus -value attribute
```

- ▼ For information about the cluster

```
# haclus -display
```

## Querying Status

- ▼ For the status of all service groups in the cluster, including resources

```
# hastatus
```

- ▼ For the status of a particular service group, including its resources

```
# hastatus -group service_group [-group service_group]...
```

- ▼ To display the status in tabular format of all systems and a specific group and resources (or all service groups if no group is specified)

```
# hastatus [-sound] [-group service_group]
```

The `-sound` option enables a bell to ring each time a resource faults.

- ▼ For the status of cluster faults, including faulted service groups, resources, systems, links, and agents

```
# hastatus -summary
```

---

**Note** Unless executed with the `-summary` option, `hastatus` continues to produce output of online state transitions until you interrupt it with the command CTRL+C.

---



## Querying Log Data Files (LDFs)

Log data files (LDFs) contain data regarding messages written to a corresponding English language file. Typically, for each English file there is a corresponding LDF.

### ▼ To display the hamsg usage list

```
# hamsg -help
```

### ▼ To display the list of LDFs available on the current system

```
# hamsg -list
```

### ▼ To display general LDF data

```
#> hamsg -info [-path path_name] LDF
```

The option `-path` specifies where `hamsg` looks for the specified LDF. If not specified, `hamsg` looks for files in the default directory `/var/VRTSvc/ldf`.

### ▼ To display specific LDF data

```
# hamsg [-any] [-tag A|B|C|D|E] [-otype VCS|RES|GRP|SYS|AGT]
        [-oname object_name] [-msgid message_ID] [-path path_name]
        [-lang language] LDF
```

The option `-any` specifies `hamsg` return messages matching any of the specified query options.

The option `-tag` specifies `hamsg` return messages matching the specified tag.

The option `-otype` specifies `hamsg` return messages matching the specified object type:

VCS = general VCS messages

RES = resource

GRP = service group

SYS = system

AGT = agent

The option `-oname` specifies `hamsg` return messages matching the specified object name.

The option `-msgid` specifies `hamsg` return messages matching the specified message ID.

The option `-path` specifies where `hamsg` looks for the specified LDF. If not specified, `hamsg` looks for files in the default directory `/var/VRTSvcs/ldf`.

The option `-lang` specifies the language in which to display messages.



## Conditional Statements

Some query commands include an option for conditional statements. Conditional statements take three forms:

Attribute=Value (the attribute equals the value)

Attribute!=Value (the attribute does not equal the value)

Attribute=~Value (the value is the prefix of the attribute, for example a query for the state of a resource = ~FAULTED would return all resources whose state begins with FAULTED.)

Multiple conditional statements can be used and imply AND logic.

---

**Note** You can only query attribute-value pairs displayed in the output of command `hagr -display` ([page 93](#)).

---

▼ **For a list of service groups whose values match a conditional statement**

```
# hagr -list [conditional_statement]
```

If no conditional statement is specified, all service groups in the cluster are listed.

▼ **For a list of resources whose values match a conditional statement**

```
# hares -list [conditional_statement]
```

If no conditional statement is specified, all resources in the cluster are listed.

▼ **For a list of agents whose values match a conditional statement**

```
# haagnt -list [conditional_statement]
```

If no conditional statement is specified, all agents in the cluster are listed.

# Basic Administrative Operations

## Administering Service Groups

▼ **To start a service group and bring its resources online**

```
# hagrpsvc -online service_group -sys system
```

▼ **To stop a service group and take its resources offline**

```
# hagrpsvc -offline service_group -sys system
```

▼ **To switch a service group from one system to another**

```
# hagrpsvc -switch service_group -to system
```

The `-switch` option is valid for failover groups only.

A service group can be switched only if it is fully or partially online.

▼ **To freeze a service group (disable onlining, offlining, and failover)**

```
# hagrpsvc -freeze service_group [-persistent]
```

The option `-persistent` enables the freeze to be “remembered” when the cluster is rebooted.

▼ **To thaw a service group (reenable onlining, offlining, and failover)**

```
# hagrpsvc -unfreeze service_group [-persistent]
```

▼ **To enable a service group**

```
# hagrpsvc -enable service_group [-sys system]
```

A group can be brought online only if it is enabled.

▼ **To disable a service group**

```
# hagrpsvc -disable service_group [-sys system]
```

A group cannot be brought online or switched if it is disabled.

▼ **To enable all resources in a service group**

```
# hagrpsvc -enableresources service_group
```



**▼ To disable all resources in a service group**

```
# hagr -disableresources service_group
```

Agents do not monitor group resources if resources are disabled.

**▼ To clear faulted, non-persistent resources in a service group**

```
# hagr -clear [service_group] -sys [system]
```

Clearing a resource automatically initiates the online process previously blocked while waiting for the resource to become clear.

- If *system* is specified, all faulted, non-persistent resources are cleared from that system only.
- If *system* is not specified, the service group is cleared on all systems in the group's SystemList in which at least one non-persistent resource has faulted.

## Administering Resources

**▼ To bring a resource online**

```
# hares -online resource -sys system
```

**▼ To take a resource offline**

```
# hares -offline resource -sys system
```

**▼ To take a resource offline and propagate the command to its children**

```
# hares -offprop resource -sys system
```

Similar to the service group `-offline` command, this command signals that its children should be taken offline. This action continues to the “leaves” of the resource's subtree.

**▼ To prompt a resource's agent to immediately monitor the resource on a particular system**

```
# hares -probe resource -sys system
```

Though the command may return immediately, the monitoring process may not be completed by the time the command returns.

### ▼ To clear a resource

Initiate a state change from `RESOURCE_FAULTED` to `RESOURCE_OFFLINE`:

```
# hares -clear resource [-sys system]
```

Clearing a resource automatically initiates the online process previously blocked while waiting for the resource to become clear. If *system* is not specified, the fault is cleared on each system in the service group's `SystemList` attribute. (See also the service group command to clear faulted, non-persistent resources, `hagrps -clear`, on [page 100](#).)

This command clears the resource's parents automatically. Persistent resources whose static attribute `Operations` is defined as `None` cannot be cleared with this command and must be physically attended to, such as replacing a raw disk. The agent then updates the status automatically.

## Administering Systems

### ▼ To force a system to start while in `ADMIN_WAIT`

```
# hasys -force system
```

This command overwrites the configuration on systems running in the cluster. Before using it, verify that the current VCS configuration is valid.

### ▼ To modify a system's attributes

```
# hasys -modify modify_options
```

Some attributes are internal to VCS and cannot be modified. For details on system attributes, see "[The -modify Option](#)" on [page 103](#).

### ▼ To display the value of a system's node ID as defined in the file `/etc/llttab`

```
# hasys -nodeid node_ID
```



▼ **To freeze a system (prevent groups from being brought online or switched on the system)**

```
# hasys -freeze [-persistent] [-evacuate] system
```

The option `-persistent` enables the freeze to be “remembered” when the cluster is rebooted. Note that the cluster configuration must be in read/write mode and must be saved to disk (dumped) to enable the freeze to be remembered.

The option `-evacuate` fails over the system’s active service groups to another system in the cluster before the freeze is enabled.

▼ **To thaw or “unfreeze” a frozen system (reenable onlining and switching of service groups)**

```
# hasys -unfreeze [-persistent] system
```

## Administering Clusters

▼ **To modify a cluster attribute**

```
# haclus [-help [-modify]]
```



## Basic Configuration Operations

Commands listed in the following sections permanently affect the configuration of the cluster. If the cluster is brought down with the command `hastop -all` or made read-only, the `main.cf` file and other configuration files written to disk reflect the updates.

### Specifying Values Preceded by a Dash (-)

When specifying values in a command-line syntax, you must prefix values beginning with a dash (-) with a percentage sign (%). If a value begins with a percentage sign, you must prefix it with another percentage sign. (The initial percentage sign is stripped by HAD and does not appear in the configuration file.)

### The -modify Option

Most configuration changes are made using the `-modify` options of the commands `haclus`, `hagrp`, `hares`, `hasys`, and `hatype`. Specifically, the `-modify` option of these commands changes the attribute values stored in the VCS configuration file. By default, all attributes are global, meaning that the value of the attribute is the same for all systems.

---

**Note** VCS must be in read/write mode before you can change the configuration. See [page 89](#) for instructions.

---



## Localizing Attributes for Resource Types

Non-static attributes of resource types may be localized, which enables you to specify a separate value for different systems. These attributes are localized on a per-resource basis. For example, to localize the attribute *attribute\_name* for *resource* only, type:

```
# hares -local resource attribute_name
```

Localizing the attribute means that for each system there is a value assigned in the service group's SystemList attribute. The original value is the same for each system, and it is the same as the original global value; however, values may be modified at the system level, as described in the following table.

Dimension	Task and Command
scalar	Replace a value: -modify [object] <i>attribute_name</i> <i>value</i> [-sys <i>system</i> ]
vector	Replace list of values: -modify [object] <i>attribute_name</i> <i>value</i> [-sys <i>system</i> ]
	Add list of values to existing list: -modify [object] <i>attribute_name</i> -add <i>value</i> [-sys <i>system</i> ]
	Update list with user-supplied values: -modify [object] <i>attribute_name</i> -update <i>entry_value</i> ... [-sys <i>system</i> ]
	Delete all values in list (you cannot delete an individual element of a vector): -modify [object] <i>attribute_name</i> -delete -keys [-sys <i>system</i> ]
keylist	Replace list of keys (duplicate keys not allowed): -modify [object] <i>attribute_name</i> <i>value</i> ... [-sys <i>system</i> ]
	Add keys to list (duplicate keys not allowed): -modify [object] <i>attribute_name</i> -add <i>value</i> ... [-sys <i>system</i> ]
	Delete user-supplied keys from list: -modify [object] <i>attribute_name</i> -delete <i>key</i> ... [-sys <i>system</i> ]
	Delete all keys from list: -modify [object] <i>attribute_name</i> -delete -keys [-sys <i>system</i> ]



Dimension	Task and Command
association	Replace list of key-value pairs (duplicate keys not allowed): <code>-modify [object] attribute_name value ... [-sys system]</code>
	Add user-supplied list of key-value pairs to existing list (duplicate keys not allowed): <code>-modify [object] attribute_name -add value ... [-sys system]</code>
	Replace value of each key with user-supplied value: <code>-modify [object] attribute_name -update key value ... [-sys system]</code>
	Delete a key-value pair identified by user-supplied key: <code>-modify [object] attribute_name -delete key ... [-sys system]</code>
	Delete all key-value pairs from association: <code>-modify [object] attribute_name -delete -keys [-sys system]</code> <b>Note</b> If multiple values are specified and if one is invalid, VCS returns an error for the invalid value, but continues to process the others. In the following example, if sysb is part of the attribute SystemList, but sysa is not, sysb is deleted and an error message is sent to the log regarding sysa. <code>hagr -modify group1 SystemList -delete sysa sysb [-sys system]</code>



## Adding Service Groups

### ▼ To add a service group to your cluster

```
# hagrps -add service_group
```

The variable *service\_group* must be unique among all service groups defined in the cluster.

This command initializes a service group that is ready to contain various resources. To employ the group properly, you must populate its SystemList attribute to define the systems on which the group may be brought online and taken offline. (A system list is an association of names and integers that represent priority values.)

## Modifying Service Group Attributes

### ▼ To modify a service group attribute

```
# hagrps -modify service_group attribute value
```

The variable *value* represents:

```
system_name1 priority system_name2 priority2
```

During a failover (with the attribute FailOver Policy set to Priority), faulted applications fail over to the system with lowest number designated in the SystemList association. Populating the system list is a way to give “hints” to HAD regarding which machine in a balanced cluster is best equipped to handle a failover.

For example, to populate the system list of service group groupx with Systems A and B, type:

```
# hagrps -modify groupx SystemList -add SystemA 1 SystemB 2
```

Similarly, to populate the AutoStartList attribute of a service group, type:

```
# hagrps -modify groupx AutoStartList SystemA SystemB
```

You may also define a service group as parallel. To set the Parallel attribute to 1, type the following command. (Note that the default for this attribute is 0, which designates the service group as a failover group.):

```
# hagr -modify groupx Parallel 1
```

This attribute cannot be modified if resources have already been added to the service group.

## Additional Considerations for Modifying Service Group Attributes

You can modify the attributes SystemList, AutoStartList, and Parallel only by using the command `hagr -modify`. You cannot modify attributes created by the system, such as the state of the service group. If you are modifying a service group from the command line, the VCS server immediately updates the configuration of the group's resources accordingly.

For example, suppose you originally defined the SystemList of service group groupx as SystemA and SystemB. Then after the cluster was brought up you added a new system to the list:

```
# hagr -modify groupx SystemList -add SystemC 3
```

The SystemList for groupx changes to SystemA, SystemB, SystemC, and an entry for `systemC` is created in the group's resource attributes, which are stored on a per-system basis. These attributes include information regarding the state of the resource on a particular system.

Next, suppose you made the following modification:

```
# hagr -modify groupx SystemList SystemA 1 SystemC 3 SystemD 4
```

Using the option `-modify` without other options erases the existing data and replaces it with new data. Therefore, after making the change above, the new SystemList becomes SystemA=1, SystemC=3, SystemD=4. SystemB is deleted from the system list, and each entry for SystemB in local attributes is removed.

## More About Modifying the SystemList Attribute

You can modify the SystemList attribute only with the commands `-modify`, `-add`, `-update`, `-delete`, or `-delete -keys`. If you modify SystemList using the command `hagr -modify` without other options (such as `-add` or `-update`), the service groups must first be taken offline on the systems being modified. *The modification fails if a service group is not offline completely.*



If you modify SystemList using the command `hagrp -modify` with the options `-delete` or `-delete -keys`, any system to be deleted that is not offline is not removed, but deleting or modifying the offline systems proceeds normally.

If you modify SystemList to add a system that has not been defined by the command `hasys -add`, the system is not added, but adding other valid systems proceeds normally.

## Adding Resources

### ▼ To add a resource

```
# hares -add resource resource_type service_group
```

This command creates a new resource, *resource*, which must be a unique name throughout the cluster, regardless of where it resides physically or in which service group it is placed. The resource type is *resource\_type*, which must be defined in the configuration language. The resource belongs to the group *service\_group*.

When new resources are created, all non-static attributes of the resource's type, plus their default values, are copied to the new resource. Three attributes are also created by the system and added to the resource:

- ◆ **Critical** (default = 1). If the resource or any of its children faults while online, the entire service group is marked “faulted” and failover occurs.
- ◆ **AutoStart** (default = 1). If the resource is set to AutoStart, it is brought online in response to a service group command. All resources designated as AutoStart=1 must be online for the service group to be considered online. (This attribute is unrelated to AutoStart attributes for service groups.)
- ◆ **Enabled**. If the resource is set to Enabled, the agent for the resource's type manages the resource. The default is 1 for resources defined in the configuration file `main.cf`, 0 for resources added on the command line.

---

**Note** Adding resources on the command line requires several steps, and the agent must be prevented from managing the resource until the steps are completed. For resources defined in the configuration file, the steps are completed before the agent is started.

---

## Modifying Attributes of a New Resource

### ▼ To modify a new resource

```
# hares -modify resource attribute value
```

The variable *value* depends on the type of attribute being created.

### ▼ To set a new resource's Enabled attribute to 1

```
# hares -modify resourceA Enabled 1
```

The resource's agent is started on a system when its Enabled attribute is set to 1 on that system. Specifically, the VCS engine begins to monitor the resource for faults. Agent monitoring is disabled if the Enabled attribute is reset to 0.

## Additional Considerations for Modifying Attributes

Resource names must be unique throughout the cluster and you cannot modify resource attributes defined by the system, such as the resource state.

## Linking Resources

### ▼ To specify a dependency relationship, or “link,” between two resources

```
# hares -link parent_resource child_resource
```

The variable *parent\_resource* depends on *child\_resource* being online before going online itself. Conversely, *parent\_resource* must take itself offline before *child\_resource* goes offline.

For example, before an IP address can be configured, its associated NIC must be available, so for resources IP1 of type IP and NIC1 of type NIC, specify the dependency as:

```
# hares -link IP1 NIC1
```

## Additional Considerations for Linking Resources

A resource can have an unlimited number of parents and children. When linking resources, the parent cannot be a resource whose Operations attribute is equal to None or OnOnly. Specifically, these are resources that cannot be brought online or taken offline by an agent (None), or can only be brought online by an agent (OnOnly).

Loop cycles are automatically prohibited by the VCS engine. You cannot specify a resource link between resources of different service groups.



## Deleting and Unlinking Service Groups and Resources

### ▼ To delete a service group

```
# hagrps -delete service_group
```

### ▼ To unlink service groups

```
# hagrps -unlink parent_group child_group
```

### ▼ To delete a resource

```
# hares -delete resource
```

Note that deleting a resource won't take offline the object being monitored by the resource. The object remains online, outside the control and monitoring of VCS.

### ▼ To unlink resources

```
# hares -unlink parent_resource child_resource
```

---

**Note** You can unlink service groups and resources at any time. You cannot delete a service group until all of its resources are deleted.

---

## Adding, Deleting, and Modifying Resource Types

After creating a resource type, use the command `haattr` to add its attributes (see “[Modifying Attributes of a New Resource](#)” on page 109). By default, resource type information is stored in the `types.cf` configuration file.

### ▼ To add a resource type

```
# hatype -add resource_type
```

### ▼ To delete a resource type

```
# hatype -delete resource_type
```

You must delete all resources of the type before deleting the resource type.

### ▼ To add or modify resource types in `main.cf` without shutting down VCS

```
# hatype -modify resource_type SourceFile "./resource_type.cf"
```

The information regarding `resource_type` is stored in the file `config/resource_type.cf`, and an include line for `resource_type.cf` is added to the `main.cf` file.

### ▼ To set the value of static resource type attributes

```
# hatype -modify ...
```

## Adding, Deleting, and Modifying Resource and Static Resource Attributes

### ▼ To add a resource attribute

```
# haattr -add resource_type attribute [value]
           [dimension] [default ...]
```

The variable `value` is a `-string` (default) or `-integer`.

The variable `dimension` is `-scalar` (default), `-keylist`, `-association`, or `-vector`.

The variable `default` is the default value of the attribute and must be compatible with the `value` and `dimension`. Note that this may include more than one item, as indicated by ellipses (`...`).

### ▼ To delete a resource attribute

```
# haattr -delete resource_type attribute
```



**▼ To add a static resource attribute**

```
# haattr -add -static resource_type attribute [value]
           [dimension] [default ...]
```

**▼ To delete a static resource attribute**

```
# haattr -delete -static resource_type attribute
```

**▼ To modify the default value of a resource attribute**

```
# haattr -default resource_type attribute new_value ...
```

The variable *new\_value* refers to the attribute's new default value.

## Starting and Stopping VCS Agents Manually

**▼ To start and stop agents manually**

```
# haagent -start agent -sys system
# haagent -stop agent -sys system
```

---

**Note** Under normal conditions, VCS agents are started and stopped automatically.

---

After issuing the commands above, a message is displayed instructing the user to look for messages in the log file. The agent log is located at `$VCS_HOME/log/agent_A.log`. See “[VCS Logging](#)” on page 293 for more information on log messages.

## Initializing Resource Type Scheduling and Priority Attributes

The following configuration shows how to initialize resource type scheduling and priority attributes through configuration files. The example shows attributes of a FileOnOff resource. (See “[Resource Attributes](#)” on page 334 for a description of each attribute cited below.)

```
type FileOnOff (
  static str AgentClass = RT
  static str AgentPriority = 10
  static str ScriptClass = RT
  static str ScriptPriority = 40
  static str ArgList[] = { PathName }
  NameRule = resource.PathName
  str PathName
)
```

## Setting Scheduling/Priority Attributes from Command Line

### ▼ To update the AgentClass

Type:

```
# hatype -modify resource_type AgentClass value
```

For example, to set the AgentClass attribute of the FileOnOff resource to RealTime, type:

```
# hatype -modify FileOnOff AgentClass "RT"
```

### ▼ To update the AgentPriority

Type:

```
# hatype -modify resource_type AgentPriority value
```

For example, to set the AgentPriority attribute of the FileOnOff resource to 10, type:

```
# hatype -modify FileOnOff AgentPriority "10"
```

### ▼ To update the ScriptClass

Type:

```
# hatype -modify resource_type ScriptClass value
```

For example, to set the ScriptClass of the FileOnOff resource to RealTime, type:

```
# hatype -modify FileOnOff ScriptClass "RT"
```

### ▼ To update the ScriptPriority

Type:

```
# hatype -modify resource_type ScriptPriority value
```

For example, to set the ScriptClass of the FileOnOff resource to RealTime, type:

```
# hatype -modify FileOnOff ScriptPriority "40"
```

---

**Note** For attributes AgentClass and AgentPriority, changes are effective immediately. For ScriptClass and ScriptPriority, changes become effective for scripts fired after the execution of the `hatype` command.

---



## Initializing Cluster Attributes in the Configuration File

You may assign values for cluster attributes while configuring the cluster. (See “[Cluster Attributes](#)” on page 360 for a description of each attribute cited below.)

Review the following sample configuration:

```
cluster vcs-india (  
  EngineClass = "RT"  
  EnginePriority = "20"  
  ProcessClass = "TS"  
  ProcessPriority = "40"  
)
```

## Setting Cluster Attributes Dynamically from Command Line

### ▼ To update the EngineClass

Type:

```
# haclus -modify EngineClass value
```

For example, to set the EngineClass attribute to RealTime, type:

```
# haclus -modify EngineClass "RT"
```

### ▼ To update the EnginePriority

Type:

```
# haclus -modify EnginePriority value
```

For example, to set the EnginePriority to 20, type:

```
# haclus -modify EnginePriority "20"
```

### ▼ To update the ProcessClass

Type:

```
# haclus -modify ProcessClass value
```

For example, to set the ProcessClass to TimeSharing, type:

```
# haclus -modify ProcessClass "TS"
```

**▼ To update the ProcessPriority**

Type:

```
# haclus -modify ProcessPriority value
```

For example, to set the ProcessPriority to 40, type:

```
# haclus -modify ProcessPriority "40"
```

---

**Note** For the attributes EngineClass and EnginePriority, changes are effective immediately. For ProcessClass and ProcessPriority changes become effective only for processes fired *after* the execution of the `haclus` command.

---





# Administering VCS from Cluster Manager (Java Console)

---

7

Cluster Manager (Java Console) offers complete administration capabilities for your cluster. Use the different views in the Java Console to monitor clusters and cluster objects, including service groups, systems, resources, and resource types. The Java Console also includes Configuration Editor, from which you can generate new configuration files (main.cf and types.cf) while VCS is offline. See the *VERITAS Cluster Server Installation Guide* for instructions on how to install Cluster Manager (Java Console).

## Getting Started

- ✓ Verify Cluster Manager (Java Console) 2.0.2 is installed.
- ✓ Verify the configuration has a user account. If a user account does not exist, you must create one. See [page 90](#). A user account is established during VCS installation that provides immediate access to Cluster Manager.
- ✓ Set the display for Cluster Manager ([page 118](#)).
- ✓ Start Cluster Manager and Configuration Editor ([page 118](#)).
- ✓ Add a cluster panel ([page 146](#)).
- ✓ Log on to a cluster ([page 148](#)).

---

**Note** Certain cluster operations are enabled or restricted depending on the privileges with which you log on to VCS. For information on specific privileges associated with VCS users, see [Chapter 5](#).

---



## Setting the Display

**Note** The UNIX version of the Cluster Manager (Java Console) requires an X-Windows desktop. *Setting the display is not required on Windows workstations.*

---

1. Type the following command to grant the system permission to display on the desktop:

```
# xhost +
```

2. Configure the shell environment variable DISPLAY on the system where Cluster Manager will be launched. For example, if using Korn shell, type the following command to display on the system “myws”:

```
# export DISPLAY=myws:0
```

## Starting Cluster Manager (Java Console) and Configuration Editor

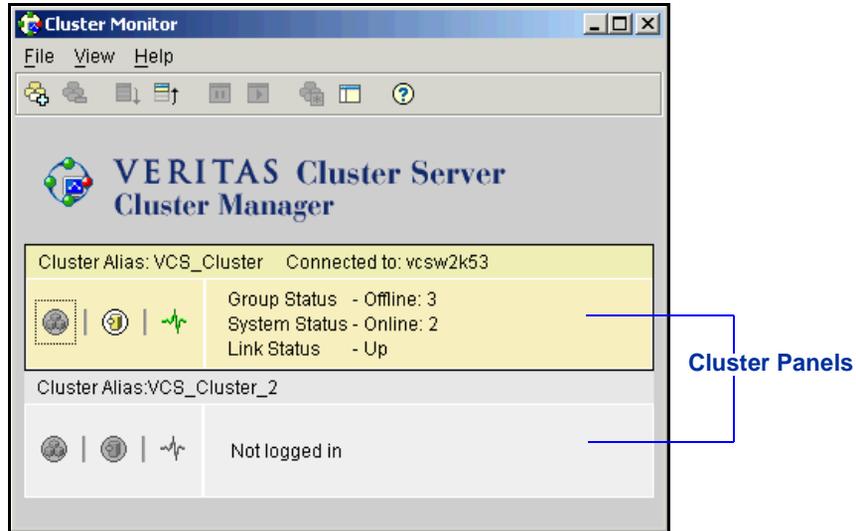
1. After establishing a user account and setting the display, type the following commands to start Cluster Manager and Configuration Editor:

```
# hagui  
# hacfed
```

2. Run `/opt/VRTSvcs/bin/hagui`.

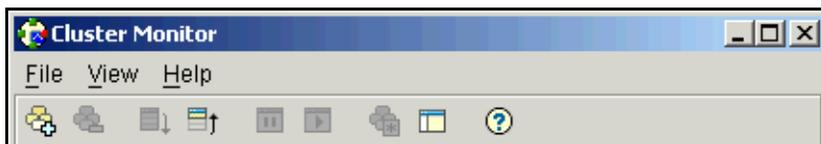
## Cluster Monitor

After starting Cluster Manager, the first window that appears is Cluster Monitor. This window includes a toolbar, menus, and one or more cluster panels displaying general information about the clusters. Use Cluster Monitor to log on to and off of a cluster, view summary information on various cluster objects, customize the display, and exit Cluster Manager.



## Cluster Monitor Toolbar and Menus

The Cluster Monitor toolbar contains nine buttons and three menus. Available operations are described below.



*From left to right:*



**New Cluster.** Adds a new cluster panel to Cluster Monitor.

*Alternative:* On the **File** menu, click **New Cluster**.



**Delete Cluster.** Removes a cluster panel from Cluster Monitor.

*Alternative:* On the **File** menu, click **Delete Cluster**.



**Expand.** Expands the Cluster Monitor view.

*Alternative:* On the **View** menu, click **Expand**.



**Collapse.** Collapses the Cluster Monitor view.

*Alternative:* On the **View** menu, click **Collapse**.



**Stop.** Pauses cluster panel scrolling.

*Alternative:* On the **View** menu, click **Stop**.



**Start.** Resumes scrolling.

*Alternative:* On the **View** menu, click **Start**.



**Login.** Log on to the cluster shown in the cluster panel.

*Alternative:* On the **View** menu, click **Login**.



**Show Explorer.** Launch the Explorer View for the cluster shown in the panel.

*Alternative:* On the **View** menu, click **Explorer**.



**Help.** Access online Help.

*Alternative:* On the **Help** menu, click **Contents**.

## Cluster Monitor Panels

The Cluster Monitor has one or more panels, each of which displays information about a cluster. The left pane displays three icons representing the state of the service groups, systems, and heartbeats. The color of the icons indicates the state of a cluster; for example:

- ◆ A flashing red slash indicates Cluster Manager failed to connect to the cluster and will attempt to connect to another system in the cluster.
- ◆ A flashing yellow slash indicates Cluster Manager is experiencing problems with the connection to the cluster.

If the system to which the console is connected goes down, a message notifies you that the connection to the cluster is lost. The console tries to connect to another system in the cluster according to the number of Failover retries set in the Connectivity Configuration window. The panel flashes until Cluster Monitor is successfully connected to a different system. If the failover is unsuccessful, a message notifies you of the failure and the panel turns gray.

Floating the cursor over an icon accesses the icon's pop-up window, which provides additional information on the specific cluster object.

The right pane of a Cluster Monitor panel displays the status (online, offline, etc.) of the various cluster objects.

To alter the connection to a cluster, right-click a panel to access a menu.

- ◆ The menu on an active panel enables you to log off a cluster.
- ◆ The menu on an inactive panel enables you to log on to a cluster, configure the cluster, and delete the cluster from Cluster Monitor.

---

**Note** An inactive panel appears gray until the user logs on and connects to the cluster.

---

Menus are enabled when the Cluster Monitor display appears in the default expanded view (see next section). If you activate a menu on a collapsed scrolling view of Cluster Monitor, the scrolling automatically stops while accessing the menu.



## Expanding and Collapsing the Cluster Monitor Display

Cluster Monitor supports two views: expanded (default) and collapsed. The expanded view shows all cluster panels. The collapsed view shows one cluster panel at a time as the panels scroll upward.

Operations enabled for the expanded view of cluster panels, such as viewing menus, are also enabled on the collapsed view after the panels stop scrolling.

### ▼ To collapse the Cluster Monitor view

On the **View** menu, click **Collapse**.

*or*

Click **Collapse** on the Cluster Monitor toolbar.

### ▼ To expand the Cluster Monitor view

On the **View** menu, click **Expand**.

*or*

Click **Expand** on the Cluster Monitor toolbar.

### ▼ To pause a scrolling cluster panel

Click the cluster panel.

*or*

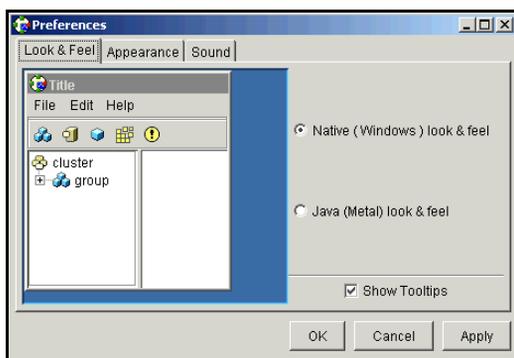
Click **Stop** on the Cluster Monitor toolbar.

## Customizing the Cluster Manager Display

Use Cluster Monitor to customize the font, color, and sound preferences of Cluster Manager.

### Selecting the Look and Feel

1. On the Cluster Monitor **File** menu, click **Preferences**.
2. Click the **Look & Feel** tab.

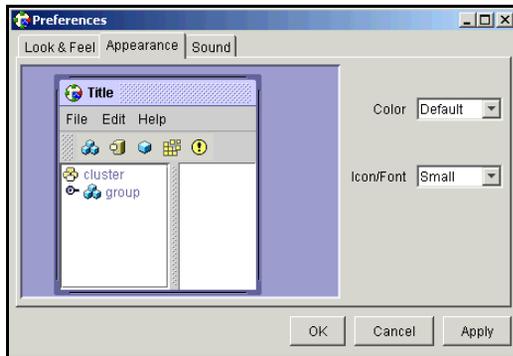


3. Click the **Native (Motif) look & feel** or the **Java (Metal) look & feel** option.
4. Click the **Show Tooltips** check box to enable ToolTips.
5. Click **Apply** to view changes.
6. Click **OK**.



## Selecting Fonts and Colors

1. On the Cluster Monitor **File** menu, click **Preferences**.
2. Click the **Appearance** tab.

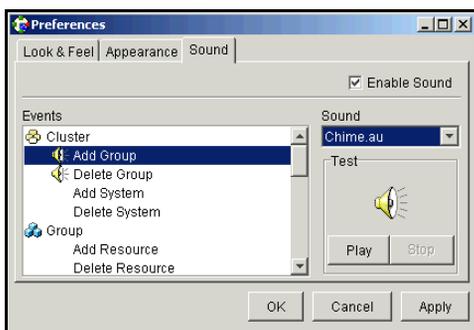


3. Click the desired color from the **Color** drop-down list box. The **Color** drop-down list box is not available for the **Native (Windows) look & feel**.
4. Click an icon and font size from the **Icon/Font** drop-down list box. This selection will not change the font size for the **Native (Windows) look & feel**.
5. Click **OK**.

## Selecting Sound Preferences

**Note** The sound feature of the Cluster Manager (Java Console) is disabled on systems for which there are no audio devices configured.

1. On the Cluster Monitor **File** menu, click **Preferences**.
2. Click the **Sound** tab.

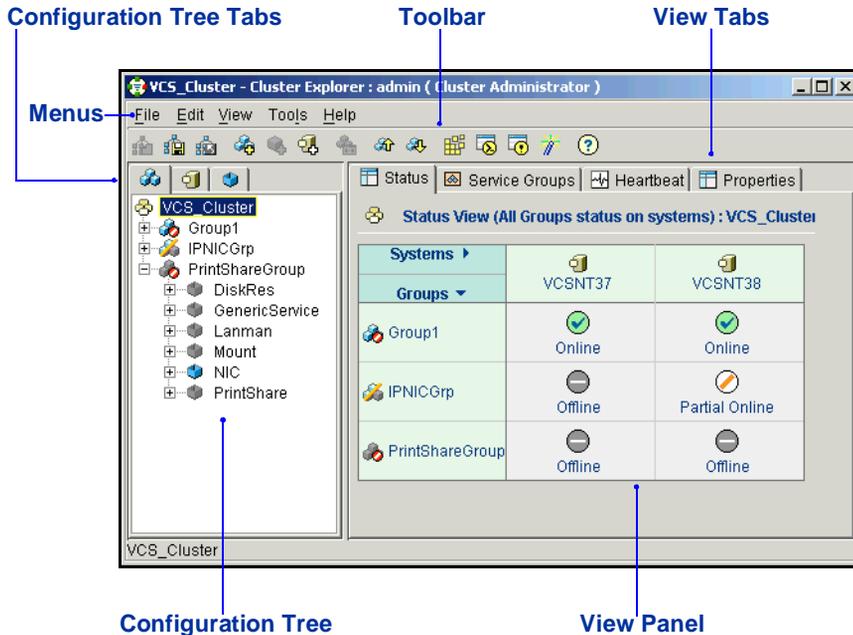


3. Click the **Enable Sound** check box to associate sound with specific events. To disable sound, clear the **Enable Sound** check box and proceed to [step 9](#).
4. Click an event from the **Events** tree.
5. Click a sound from the **Sound** drop-down list box.
6. To test the selected sound, click **Play**.
7. Click **Apply**.
8. Repeat [step 4](#) through [step 7](#) to enable sound for other events.
9. After you have made your final selection, click **OK**.



## Cluster Explorer

Cluster Explorer is the main window for cluster administration. From this window you can view the status of cluster objects and perform various operations.



The display is divided into three panes. The top pane includes a toolbar that enables you to perform frequently used operations quickly. The left pane contains a configuration tree with three tabs: Service Groups, Systems, and Resource Types. The right pane contains a view panel that displays details of the object selected in the configuration tree.

Certain colors and conventions in the display indicate the state of the cluster.

- ◆ A red slash indicates at least one object is faulted and no objects are online.
- ◆ A yellow slash indicates some objects are faulted and some objects are online.
- ◆ A gray or "dimmed" object is offline.
- ◆ A "locked" icon indicates the object is frozen.

## ▼ To access Cluster Explorer

1. Log on to the cluster ([page 148](#)).
2. Click anywhere in the active Cluster Monitor panel.

*or*

Right-click the selected Cluster Monitor panel and click **Explorer View** on the menu.

## The Cluster Explorer Toolbar

The Cluster Explorer toolbar contains 14 buttons, each of which performs a specific operation. Note that some buttons may be disabled, depending on the privileges with which you logged on to the cluster.



*From left to right:*



**Open Configuration.** Modifies a read-only configuration to a read-write file. This enables you to modify the configuration.



**Save Configuration.** Writes the configuration to disk.



**Close Configuration.** Writes the configuration to disk as a read-only file.



**Add Service Group.** Displays the Add Service Group dialog box.



**Add Resource.** Displays the Add Resource dialog box.



**Add System.** Displays the Add System dialog box.



**Manage Systems for a Service Group.** Displays the System Manager dialog box.



**Online Service Group.** Displays the Online Service Group dialog box.





**Offline Service Group.** Displays the Offline Service Group dialog box.



**Show Command Center.** Enables you to build and execute VCS commands.



**Show Shell Command Window.** Enables you to launch a non-interactive shell command on cluster systems, and to view the results on a per-system basis.



**Show the Log Desk.** Displays messages received from the VCS engine, HAD, and the commands issued from the console.



**Launch Configuration Wizard.** Enables you to create VCS service groups.



**Show Cluster Explorer Help.** Enables you to access online help.

## The Cluster Explorer Configuration Tree

The Cluster Explorer configuration tree is a tabbed display of cluster objects.

- ◆ The Service Groups tab displays the service groups in the cluster. Expand each service group to view the group's resource types and resources.
- ◆ The Systems tab displays the systems in the cluster.
- ◆ The Types tab displays the resource types in the cluster.

## The Cluster Explorer View Panel

The right pane of the Cluster Explorer includes a view panel with detailed information about the object selected in the configuration tree. The information is presented in tabular or graphical format. Use the tabs in the view panel to access a particular view; the tabs are activated based on the cluster object selected in the configuration tree.

- ◆ Click any cluster object in the configuration tree to access the Status View and Properties View.
- ◆ Click a cluster in the configuration tree to access the Service Group View and Heartbeat View.
- ◆ Click a service group in the configuration tree to access the Resource View.

### ▼ To create a “tear-off” view

On the **View** menu, click **Tear Off** and click the appropriate view from the menu.

*or*

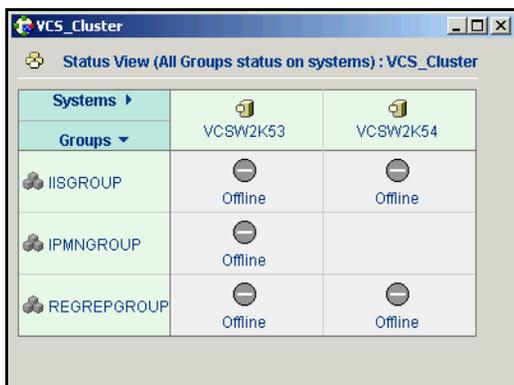
Right-click the cluster object in the configuration tree, click **View**, and click the appropriate view from the menu.



## The Status View

The Status View summarizes the state of the object selected in the configuration tree. Use this view to monitor the status of a cluster, system, service group, resource type, and resource.

If a cluster is selected in the configuration tree, the Status View uses icons to display the state of the cluster. Float the cursor over an icon to open a pop-up window with information about the cluster object.



### ▼ To access the Status View

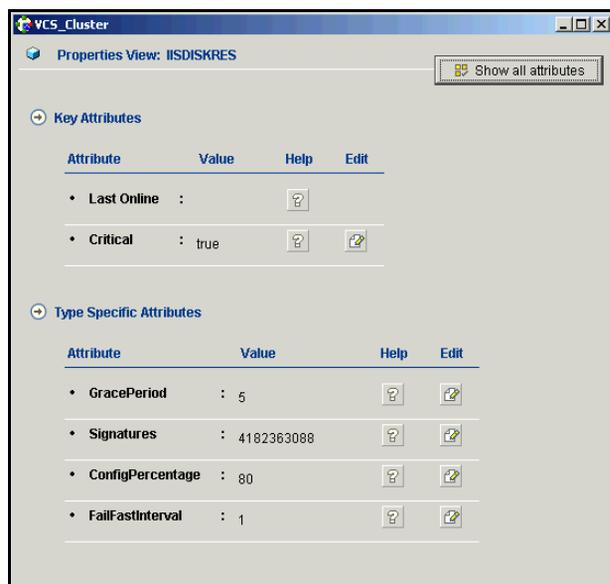
1. From **Cluster Explorer**, click a cluster in the configuration tree.
2. In the view panel, click the **Status** tab.

## The Properties View

The Properties View displays the attributes of cluster objects. These attributes define the scope and parameters of a cluster and its components. For example, the value of a service group's SystemList attribute specifies the systems on which the group is configured, and the priority of each system within the group. The view also enables you to edit attribute values (page 199).

By default, this view displays key attributes of the object selected in the configuration tree. The Properties View for a resource displays key attributes of the resource and attributes specific to the resource types. To view all attributes associated with the selected cluster object, click **Show all attributes**.

For more information about an attribute, click the attribute name or the icon in the Help column of the Key Attributes or Type Specific Attributes table. For a complete list of VCS attributes, including their type, dimension, scope, and definition, see [Appendix B](#).



### ▼ To access the Properties View

1. From **Cluster Explorer**, click a cluster object in the configuration tree.
2. In the view panel, click the **Properties** tab.



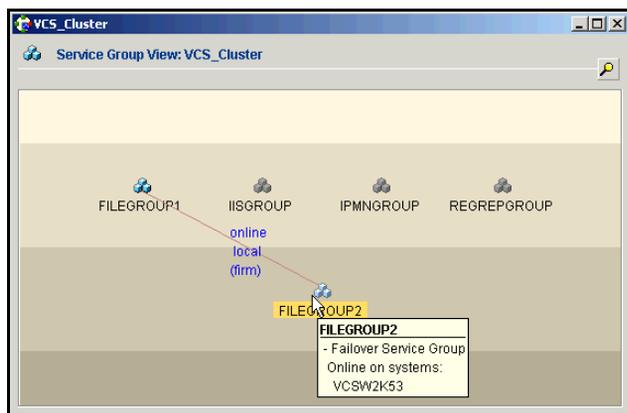
## The Service Group View

The Service Group View displays the service groups in a cluster. Use the graph in this view to monitor the dependencies between service groups. In the graph, the line between two service groups represents a dependency, or “parent-child” relationship. Service group dependencies specify the order in which service groups are brought online and taken offline. (See [Chapter 4](#) for details.)

- ◆ A service group that depends on other service groups is a parent group. The graph links a parent group icon to a child group icon below it.
- ◆ A service group on which the other service groups depend is a child group. The graph links a child group icon to a parent group icon above it.
- ◆ A service group can function as a parent and a child.

The color of the link between service groups indicates different types of dependencies.

- ◆ A blue link indicates a soft dependency. The child group must be online before bringing the parent group online, but the parent group is not automatically taken offline when the child faults.
- ◆ A red link indicates a firm dependency. The child group must be online before bringing the parent group online, and the parent group is automatically taken offline when the child faults. When the child is brought online on another system, the parent group is brought online on any system other than the system on which it was taken offline.



### ▼ To access the Service Group View

1. From **Cluster Explorer**, click a cluster in the configuration tree.
2. In the view panel, click the **Service Groups** tab.



## The Resource View

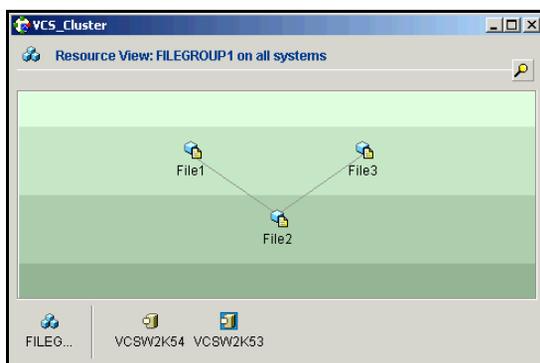
The Resource View displays the resources in a service group. Use the graph and pop-up windows in this view to monitor the dependencies between resources and the status of the service group on all or individual systems in a cluster.

Similar to the Service Group View, the line between two resources represents a dependency, or “parent-child” relationship. Resource dependencies specify the order in which resources are brought online and taken offline. (See “[Resource Dependencies](#)” on page 14 for details.)

- ◆ A resource that depends on other resources is a parent resource. The graph links a parent resource icon to a child resource icon below it. Root resources (resources without parents) are displayed in the top row.
- ◆ A resource on which the other resources depend is a child resource. The graph links a child resource icon to a parent resource icon above it.
- ◆ A resource can function as a parent and a child.

Float the cursor over a resource icon to display pop-up windows with information about the type, state, and key attributes of the resource. The state of the resource can reflect the state across all systems (global), or the state on a specified system (local).

In the bottom pane of the Resource View, float the cursor over the system and service group icons to display pop-up windows with information about the service group status on all or individual systems in a cluster. Click a system icon to view the resource graph of the service group on the system. Click the service group icon to view the resource graph on all systems in the cluster.



### ▼ To access the Resource View

1. From **Cluster Explorer**, click the **Service Groups** tab in the configuration tree.
2. Click a service group in the configuration tree.
3. In the view panel, click the **Resources** tab.

## The Zoom Panel for Resource and Service Group Views

Both the Resource and Service Group Views include a navigator tool to zoom in or out of their graphs. This tool is useful for large configurations that are difficult to view from the standard view panel. Click the magnifying glass icon in the top right corner to open the zoom panel.

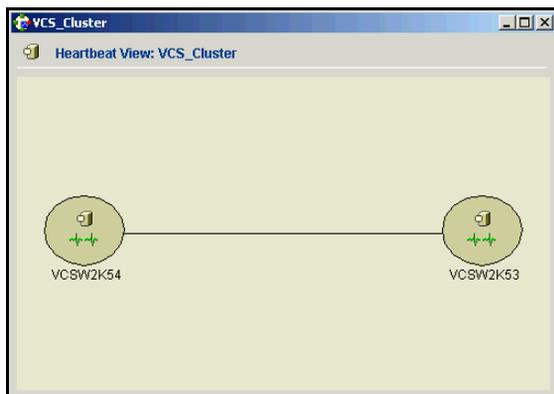


- ◆ To move the view to the left or right, click a distance (in pixels) from the drop-down list box between the hand icons. Click the left or right hand icon to move the view in the desired direction.
- ◆ To shrink or enlarge the view, click a size factor from the drop-down list box between the magnifying glass icons. Click the - or + magnifying glass icon to modify the size of the view.
- ◆ To view a segment of the graph, place the cursor in the box to the right of the + magnifying glass icon. Use the red outline in this box to encompass the appropriate segment of the graph. Click the newly outlined area to view the segment.
- ◆ To return to the original view, click the magnifying glass icon labeled 1.

## The Heartbeat View

The Heartbeat View displays the status of system connections in a cluster. Use this view to monitor the system links and disk group heartbeats.

Float the cursor over a system icon to display pop-up windows with information on the links and disk group heartbeats. If a system in the cluster is experiencing a problem connecting to other systems, the system icon changes its appearance to indicate the link or disk heartbeat is down. In this situation, a jeopardy warning may also appear in the pop-up window for this system.



### ▼ To access the Heartbeat View

1. From **Cluster Explorer**, click a cluster in the configuration tree.
2. In the view panel, click the **Heartbeat** tab.

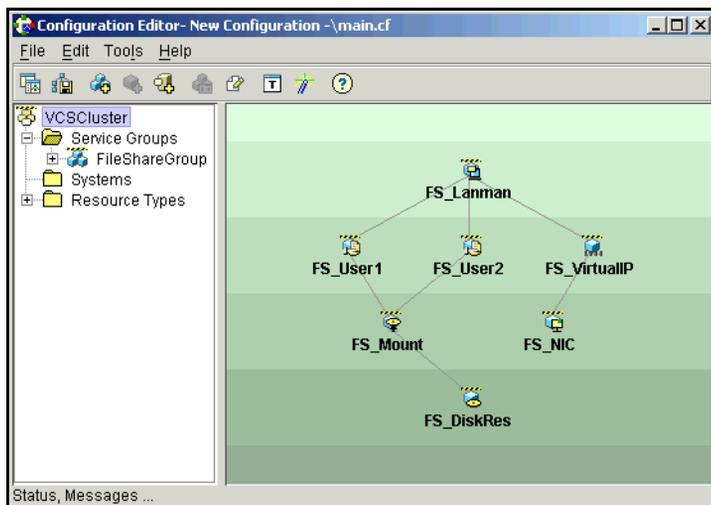


## Configuration Editor

Configuration Editor enables you to create and edit existing configuration files (main.cf and types.cf) on the local system while VCS is offline. Many of the operations supported in Configuration Editor are also supported in Cluster Explorer.

- ◆ Use Configuration Editor to modify a configuration not currently connected to a particular host. This tool is useful in situations where the user needs to create a configuration for one platform while working on another.
- ◆ Use Cluster Explorer to modify a configuration while connected to a specific host. In this situation, you cannot use the templates for a platform other than the host platform.

Configuration Editor includes a configuration wizard and a set of predefined templates for configuring standard applications as service groups. Resource dependency graphs are provided for each template. *These templates and the hacf utility must reside on the local system.* Use Configuration Editor to add service groups, resources, and systems to your existing configuration, and to specify system details for a service group.



The top pane of the Configuration Editor window contains the Configuration Editor toolbar. The left pane contains the configuration tree, a dynamic hierarchical display of the cluster, including its service groups, the resources within the service groups, and resource types. The right pane contains a view panel that displays the resource dependency graph of the selected service group or template. Double-click an object to view or modify its attributes.

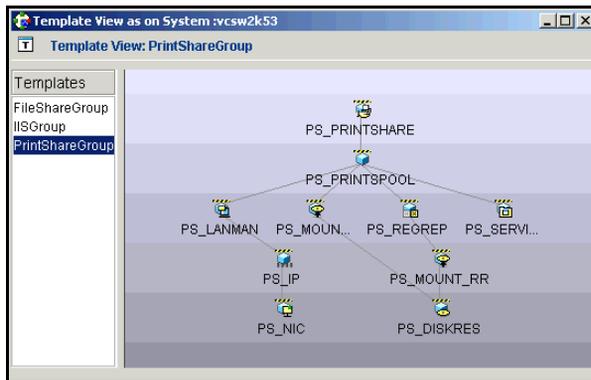
## Accessing Additional Features of the Java Console

Use Cluster Manager and Configuration Editor to access the Template View, System Manager, User Manager, Cluster Shell, Command Center, Configuration Wizard, and Log Desk.

### The Template View

The Template View displays the service group templates available in VCS. Templates are predefined service groups that define the resources, resource attributes, and dependencies within the service group. Use this view to add service groups to the cluster configuration, and copy the resources within a service group template to existing service groups.

In this window, the left pane displays the templates available on the system to which Cluster Manager is connected. The right pane displays the selected template's resource dependency graph. Template files conform to the VCS configuration language and contain the extension `.tf`.



#### ▼ To access the Template View

On the Cluster Explorer **Tools** menu, click **Templates**.

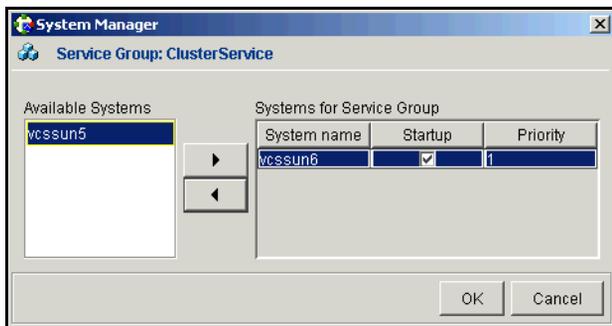
*or*

On the Configuration Editor **Tools** menu, click **Templates**.



## System Manager

From Cluster Explorer or Configuration Editor, use System Manager to add and remove systems in a service group's system list. Configuration Editor enables you to use System Manager for a configuration on the local system while VCS is offline.



### ▼ To access System Manager

From **Cluster Explorer**, click the service group in the configuration tree. On the **Tools** menu, click **System Manager**.

*or*

In the **Service Groups** tab of the Cluster Explorer configuration tree, click the service group. Click **Manage Systems for a Service Group** in the Cluster Explorer toolbar.

*or*

On the **Configuration Editor Tools** menu, click **System Manager**.

*or*

Right-click the service group in the Configuration Editor configuration tree, and click **System Manager** from the menu.

## User Manager

From Cluster Explorer, use User Manager to add and delete user profiles, change user passwords, and change user privileges. You must be logged in as Cluster Administrator to access User Manager.



### ▼ To access User Manager

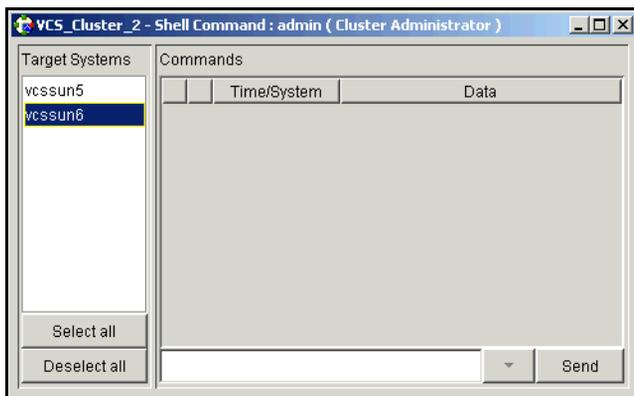
On the Cluster Explorer **File** menu, click **User Manager**.



## Cluster Shell

Cluster Shell enables you to run a non-interactive shell command on one or more cluster systems, and to view the results for each system. *Do not use Cluster Shell to run commands that require user input.*

The left pane of the window displays a list of the systems in the cluster. The right pane displays the time the command was issued, the system on which it was issued, and the output from the command as generated on the system's command line or console. The bottom pane provides a field to enter the command.



The following conditions must be met to access and run commands from this window:

- ✓ Set the cluster attribute `HacliUserLevel` to `CLUSTERADMIN`. For more information on this attribute, see [Appendix B](#).  
If the attribute `HacliUserLevel` was modified while Cluster Explorer was open, close and restart Cluster Explorer before using Cluster Shell.
- ✓ Set the user account with Cluster Administrator privileges.

---

**Note** If arguments to a command entered from this window contain a space, enclose the arguments within double quotation marks.

---

### ▼ To access Cluster Shell

On the Cluster Explorer **Tools** menu, click **Cluster Shell**.

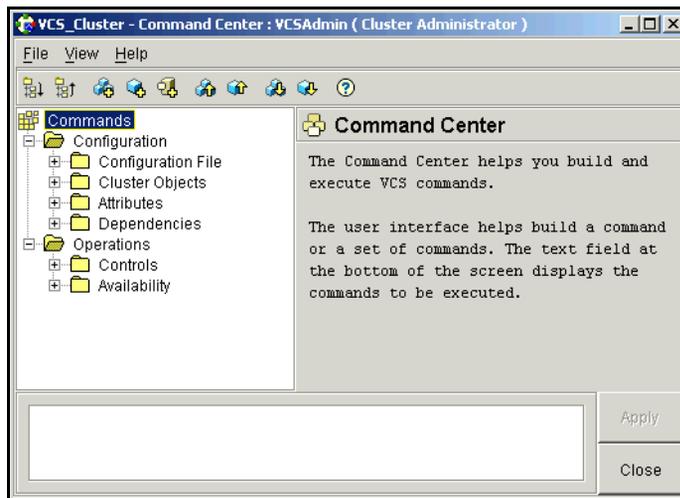
*or*

In the Cluster Explorer toolbar, click **Show Shell Command Window**.

## Command Center

Command Center enables you to build and execute VCS commands. The left pane of the window displays a commands tree of all VCS operations. The right pane displays a view panel that describes the selected command. The bottom pane displays the commands being executed.

The commands tree is organized into Configuration and Operations folders. Click the icon to the left of the Configuration or Operations folder to view its subfolders and command information in the right pane. Float the cursor over an entry in the commands tree to display a pop-up window with information about the selected command.



### ▼ To access Command Center

On the Cluster Explorer **Tools** menu, click **Command Center**.

*or*

In the Cluster Explorer toolbar, click **Show Command Center**.

*or*

From Cluster Monitor, right-click an active panel and click **Command Center** from the menu.



## The Command Center Configuration Folder

<b>Subfolder</b>	<b>Operations</b>
Configuration File	Open Configuration
	Save Configuration
	Close Configuration
Cluster Objects	Add Service Group
	Add Resource
	Add System
	Delete Service Group
	Delete Resource
	Delete System
Attributes	Modify Cluster Attributes
	Modify Service Group Attributes
	Modify Resource Attributes
	Modify System Attributes
	Modify Resource Type Attributes
Dependencies	Link Resources
	Link Service Groups
	Unlink Resources
	Unlink Service Groups



## The Command Center Operations Folder

Subfolder	Operations
Monitoring	Enable Link Monitoring (for clusters created with VCS prior to 2.0)
	Disable Link Monitoring (for clusters created with VCS prior to 2.0)
Controls	Online Service Group
	Online Resource
	Offline Service Group
	Offline Resource
	Switch Service Group
	Offprop Resource
	Probe Resource
Availability	Freeze Service Group
	Freeze System
	Unfreeze Service Group
	Unfreeze System
	Enable Service Group
	Enable Resources for Service Group
	Disable Service Group
	Disable Resources for Service Group
	Autoenable Service Group
	Clear Resource
Flush Service Group	



## Configuration Wizard

From Cluster Explorer or Configuration Editor, use Configuration Wizard to create and configure service groups in a cluster.

### ▼ To access Configuration Wizard

On the Cluster Explorer **Tools** menu, click **Configuration Wizard**.

*or*

In the Cluster Explorer toolbar, click **Launch Configuration Wizard**.

*or*

On the Configuration Editor **Tools** menu, click **Configuration Wizard**.

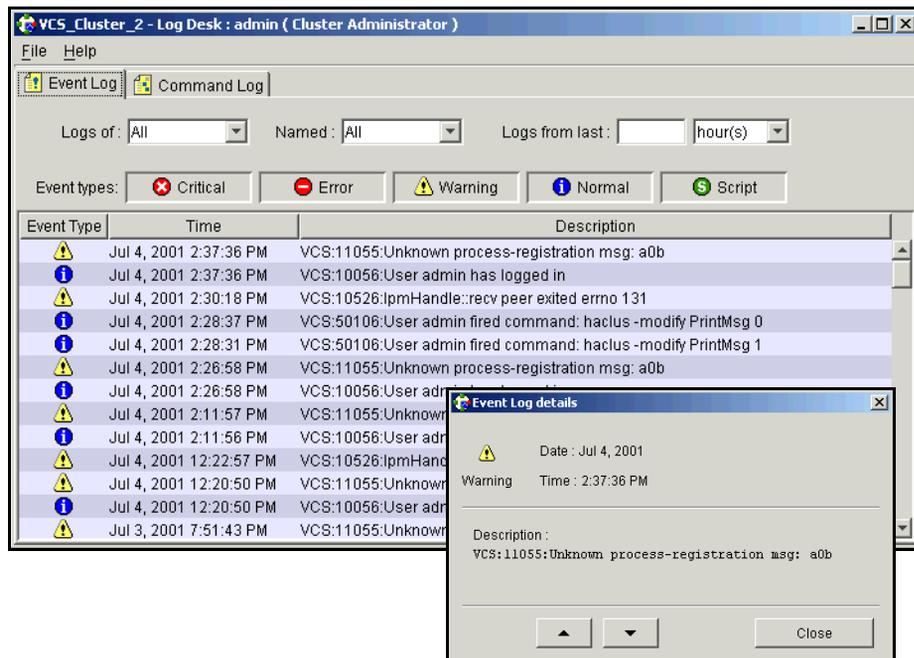
*or*

In the Configuration Editor toolbar, click **Launch Configuration Wizard**.

## Log Desk

The Log Desk displays the log messages generated by the VCS engine, HAD, and commands issued from Cluster Manager to the cluster. Click the Event Log tab to view the type, time, and details of an event. Click the Command Log tab to view the status (success or failure), time, command ID, and details of a command. The Command Log only displays commands issued in the current session.

Each Event Log message presents an icon in the first column of the table to indicate the type of message. Use this window to customize the display of messages by setting filter criteria.



### ▼ To access the Log Desk window

On the Cluster Explorer **View** menu, click **Log Desk**.

*or*

In the Cluster Explorer toolbar, click **Show the Log Desk**.

*or*

From Cluster Monitor, right-click an active panel and click **Log Desk** from the menu.



## Using Cluster Monitor

The Java Console enables you to administer a cluster by adding or reconfiguring a cluster panel in Cluster Monitor. To activate the connection of either procedure, log on to the cluster after completing the final step.

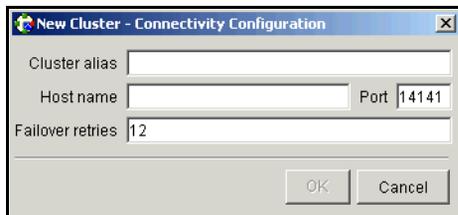
### Configuring a New Cluster Panel

1. On the Cluster Monitor **File** menu, click **New Cluster**.

*or*

In the Cluster Monitor toolbar, click **New Cluster**.

2. In the **New Cluster - Connectivity Configuration** dialog box, enter the cluster alias, host name, port number, and number of failover retries. VCS sets the default port number to 14141 and failover retries number to 12. The cluster alias is designated by the user and is not necessarily the name of the cluster recognized by the VCS engine, HAD. It appears on the local display only to differentiate the clusters being monitored.



3. Click **OK**. An inactive panel appears in Cluster Monitor.

## Configuring an Existing Cluster Panel

1. If Cluster Monitor is in the default expanded state, proceed to [step 2](#). If Cluster Monitor is in the collapsed state:  
On the **View** menu, click **Expand**.  
*or*  
In the Cluster Monitor toolbar, click **Expand**.  
*or*  
On the **View** menu, click **Stop** when an active panel appears as the view panel.  
*or*  
Click **Stop** in the Cluster Monitor toolbar when an active panel appears as the view panel.
2. Right-click the cluster panel. If the panel is inactive, proceed to [step 5](#).
3. On the menu, click **Logout**. The cluster panel becomes inactive.
4. Right-click the inactive panel.
5. On the menu, click **Configure**.
6. Enter values for the host name, port number, and the number of failover retries. VCS sets the default port number to 14141 and failover retries number to 12.
7. Click **OK**.



## Logging On to and Off of a Cluster

After you add or configure a cluster panel in Cluster Monitor, log on to a cluster to access Cluster Explorer. Use Cluster Monitor to log off a cluster when you have completed administering the cluster.

### Logging On to a Cluster

1. If Cluster Monitor is in the default expanded state, proceed to [step 2](#). If Cluster Monitor is in the collapsed state:

On the **View** menu, click **Expand**.

*or*

Click **Expand** in the Cluster Monitor toolbar.

*or*

On the **View** menu, click **Stop** when an active panel appears as the view panel.

*or*

Click **Stop** in the Cluster Monitor toolbar when an active panel appears as the view panel.

2. Click the panel that represents the cluster you want to log on to and monitor.
3. In the **Login** window, enter a valid VCS user name and password.
4. Click **OK**. The animated display shows various cluster objects, such as service groups and resources, being transferred from the server to the console. Cluster Explorer is launched automatically upon initial logon, and the icons in the cluster panel change color to indicate an active panel.

## Logging Off of a Cluster

### ▼ To log off from Cluster Monitor

1. If Cluster Monitor is in the default expanded state, proceed to [step 2](#). If Cluster Monitor is in the collapsed state:

On the **View** menu, click **Expand**.

*or*

Click **Expand** in the Cluster Monitor toolbar.

*or*

On the **View** menu, click **Stop** when an active panel appears as the view panel.

*or*

Click **Stop** in the Cluster Monitor toolbar when an active panel appears as the view panel.

2. Right-click the active panel.
3. On the menu, click **Logout**. Cluster Explorer closes and the Cluster Monitor panel becomes inactive. You may be prompted to save the configuration if any commands were executed on the cluster.

### ▼ To log off from Cluster Explorer

On the **File** menu, click **Log Out**.

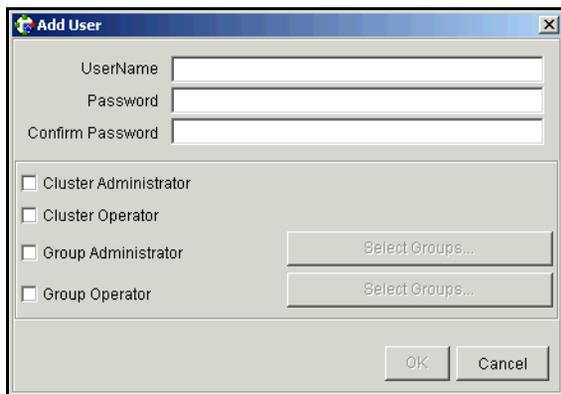


## Administering User Profiles

The Java Console enables a user with Cluster Administrator privileges to add, modify, and delete user profiles. The icon next to each user name in the User Manager dialog box indicates privileges for each user. Administrator and Operator privileges are separated into cluster and group levels.

### Adding a User

1. On the Cluster Explorer **File** menu, click **User Manager**.
2. In the **User Manager** dialog box, click **New User**.
3. In the **Add User** dialog box, enter the name of the user.



4. Enter the password.
5. Reenter the password in the **Confirm Password** field.
6. Click the appropriate check boxes to grant privileges to the user. To grant Group Administrator or Group Operator privileges, proceed to the next step. Otherwise, proceed to [step 11](#).
7. Click **Select Groups**.

8. In the **Select Groups** dialog box, click the groups for which you want to grant privileges to the user.



9. Click the right arrow to move the groups to the **Selected Groups** box.
10. Click **OK** to close the **Add Groups** dialog box.
11. Click **OK**.

## Deleting a User

1. On the Cluster Explorer **File** menu, click **User Manager**.
2. Click the user name.
3. Click **Remove User**.



## Changing a User Password

A user with Administrator, Operator, or Guest privileges can change his or her own password. You must be logged in as Cluster Administrator to access User Manager.

### ▼ To change a password as Administrator

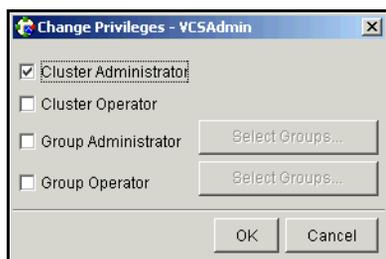
1. On the Cluster Explorer **File** menu, click **User Manager**.
2. Click the user name and click **Change Password**.
3. In the **Change Password** dialog box, enter the new password.
4. Reenter the password in the **Confirm Password** field.
5. Click **OK**.

### ▼ To change a password as Operator or Guest

1. On the Cluster Explorer **File** menu, click **Change Password**.
2. In the **Change Password** dialog box, enter the new password.
3. Reenter the password in the **Confirm Password** field.
4. Click **OK**.

## Changing User Privileges

1. On the Cluster Explorer **File** menu, click **User Manager**.
2. Click the user name and click **Change Privileges**.
3. In the **Change Privileges** dialog box, click the appropriate check boxes to grant privileges to the user. To grant Group Administrator or Group Operator privileges, proceed to [step 4](#). Otherwise, proceed to [step 7](#).



4. Click **Select Groups**.
5. On the **Available Groups** box, click the groups for which you want to grant privileges to the user.



6. Click the right arrow to move the groups to the **Selected Groups** box.
7. Click **OK** to close the **Change Privileges** dialog box.



## Administering Service Groups

The Java Console enables you to administer service groups in the cluster. Use the console to add and delete, bring online and take offline, freeze and unfreeze, link and unlink, enable and disable, autoenable, switch, and flush service groups. You can also modify the system list for a service group.

### Adding a Service Group

The Java Console provides several ways to add a service group to the systems in a cluster. Use Cluster Explorer, Command Center, Configuration Editor, or the Template View to perform this task.

#### ▼ To add a service group from Cluster Explorer

1. On the **Edit** menu, click **Add**, and click **Service Group**.

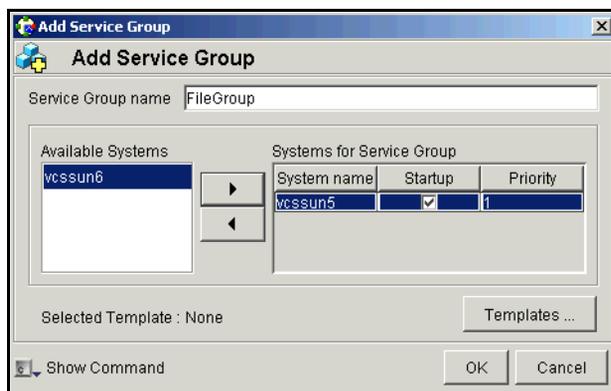
*or*

In the **Service Groups** tab of the configuration tree, right-click a cluster and click **Add Service Group** from the menu.

*or*

Click **Add Service Group** in the Cluster Explorer toolbar.

2. Enter the name of the service group.



3. In the **Available Systems** box, click the systems on which the service group will be added.

4. Click the right arrow to move the selected systems to the **Systems for Service Group** box.
5. To add a new service group based on a template, click **Templates**. Otherwise, proceed to [step 8](#).
6. Click the appropriate template name in the **Templates** box.
7. Click **OK**.
8. In the **Add Service Group** dialog box, click **Show Command** in the bottom left corner if you want to view the command associated with the service group. Click **Hide Command** to close the view of the command.
9. Click **OK**.

▼ **To add a service group from Command Center**

1. Click **Commands>Configuration>Cluster Objects>Add Service Group**.  
*or*  
Click **Add service group** in the Command Center toolbar.
2. Enter the name of the service group.
3. In the **Available Systems** box, click the systems on which the service group will be added.
4. Click the right arrow to move the selected systems to the **Systems for Service Group** box.
5. To add a new service group based on a template, click **Templates**. Otherwise, proceed to [step 8](#).
6. Click the appropriate template name in the **Templates** box.
7. Click **OK**.
8. In the **Command Center** window, click **Apply**.



▼ **To add a service group from the Template View**

1. On the Cluster Explorer **Tools** menu, click **Templates**.

*or*

On the Configuration Editor **Tools** menu, click **Templates**.

2. Right-click the **Template View** panel, and click **Add as Service Group** from the menu. This adds the service group template to the cluster configuration file without associating it with a particular system.
3. Use **System Manager** to add the service group to systems in the cluster.

▼ **To add a service group from Configuration Editor**

Use Configuration Editor to add a service group on the local system while VCS is offline. Some options in the following procedure may not be available, depending on the specifications in the configuration file.

1. Click a cluster or the Service Groups folder in the configuration tree. On the **Edit** menu click **Add**, and click **Service Group**.

*or*

Right-click a cluster or the Service Groups folder in the configuration tree, click **Add**, and click **Service Group** from the menu.

*or*

Click **Add Service Group** from the Configuration Editor toolbar.

2. In the **Add Service Group** dialog box, enter the name of the service group.
3. Click the systems on which the service group will be added.
4. Click the right arrow to move the selected systems to the Systems for Service Group box. To remove a system from the Systems for Service Group box, click the system and click the left arrow.
5. In the **Systems for Service Group** table, click the **Startup** check box to start the service group on the system automatically. (The priority number indicates the failover sequence, with 1 denoting the first system to start after a failover.)
6. To add a new service group based on a template, click **Templates**. Otherwise, proceed to [step 9](#).
7. Click the appropriate template name in the **Templates** box.

8. Click **OK**.
9. In the **Add Service Group** dialog box, click **Show Command** in the bottom left corner if you want to view the command associated with the service group. Click **Hide Command** to close the view of the command.
10. Click **OK**.

## Deleting a Service Group

Delete a service group from Cluster Explorer, Command Center, or Configuration Editor. Use Configuration Editor to delete a service group on the local system while VCS is offline.

---

**Note** You cannot delete service groups with dependencies. To delete a linked service group, you must first delete the link.

---

### ▼ To delete a service group from Cluster Explorer

1. In the **Service Groups** tab of the configuration tree, right-click the service group.  
*or*  
Click a cluster in the configuration tree, click the **Service Groups** tab, and right-click the service group icon in the view panel.
2. Click **Delete** from the menu.

### ▼ To delete a service group from Command Center

1. Click **Commands>Configuration>Cluster Objects>Delete Service Group**.
2. Click the service group.
3. Click **Apply**.



▼ **To delete a service group from Configuration Editor**

1. Click the service group in the configuration tree.
2. On the **Edit** menu, click **Delete**, and click **Service Group**.  
*or*  
Right-click the service group in the configuration tree, and click **Delete** from the menu.
3. Click **Yes** to confirm the deletion of the service group and all resources within the group.

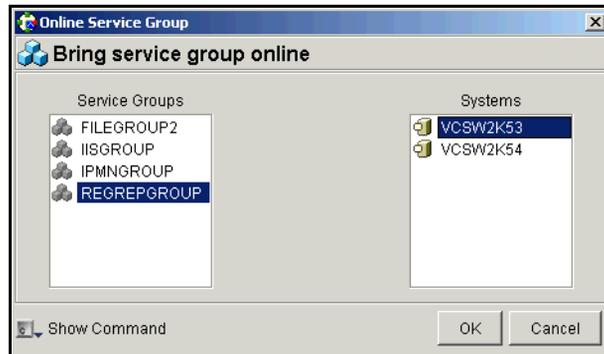
## Bringing a Service Group Online

▼ **To bring a service group online from the Cluster Explorer Configuration Tree**

1. In the **Service Groups** tab of the configuration tree, right-click the service group.  
*or*  
Click a cluster in the configuration tree, click the **Service Groups** tab, and right-click the service group icon in the view panel.
2. Click **Online**, and click the appropriate system from the menu.

▼ **To bring a service group online from the Cluster Explorer Toolbar**

1. Click **Online Service Group** in the Cluster Explorer toolbar.
2. In the **Online Service Group** dialog box, click the service group.



3. Click the system on which to bring the service group online.
4. Click **Show Command** in the bottom left corner if you want to view the command associated with the service group. Click **Hide Command** to close the view of the command
5. Click **OK**.

▼ **To bring a service group online from Command Center**

1. Click **Commands>Operations>Controls>Online Service Group**.  
*or*  
Click **Bring service group online** in the Command Center toolbar.
2. Click the service group.
3. Click the system on which to bring the service group online.
4. Click **Apply**.



## Taking a Service Group Offline

### ▼ To take a service group offline from Cluster Explorer Configuration Tree

1. In the **Service Groups** tab of the configuration tree, right-click the service group.

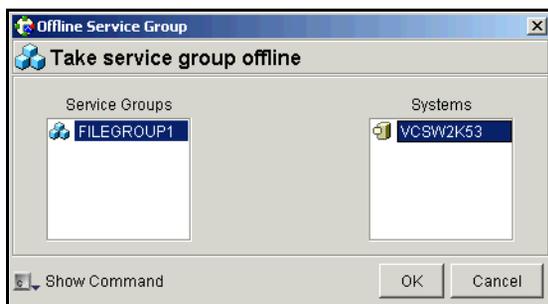
*or*

Click a cluster in the configuration tree, click the **Service Groups** tab, and right-click the service group icon in the view panel.

2. Click **Offline**, and click the appropriate system from the menu.

### ▼ To take a service group offline from the Cluster Explorer Toolbar

1. Click **Offline Service Group** in the Cluster Explorer toolbar.
2. In the **Offline Service Group** dialog box, click the service group.



3. Click the system on which to take the service group offline.
4. Click **Show Command** in the bottom left corner if you want to view the command associated with the service group. Click **Hide Command** to close the view of the command.
5. Click **OK**.

### ▼ To take a service group offline from Command Center

1. Click Commands>Operations>Controls>Offline Service Group.  
*or*  
Click **Take service group offline** in the Command Center toolbar.
2. Click the service group.
3. Click the system on which to take the service group offline.
4. Click **Apply**.

## Switching a Service Group

The process of switching a service group involves taking it offline on its current system and bringing it online on another system.

### ▼ To switch a service group from Cluster Explorer

1. In the **Service Groups** tab of the configuration tree, right-click the service group.  
*or*  
Click the cluster in the configuration tree, click the **Service Groups** tab, and right-click the service group icon in the view panel.
2. Click **Switch To**, and click the appropriate system from the menu.

### ▼ To switch a service group from Command Center

1. Click Commands>Operations>Controls>Switch Service Group.
2. Click the service group.
3. Click the system to which the service group will be switched.
4. Click **Apply**.



## Freezing a Service Group

Freeze a service group to prevent it from failing over to another system. This process stops all online and offline procedures on the service group.

### ▼ To freeze a service group from Cluster Explorer

1. In the **Service Groups** tab of the configuration tree, right-click the service group.

*or*

Click the cluster in the configuration tree, click the **Service Groups** tab, and right-click the service group icon in the view panel.

2. Click **Freeze**, and click **Temporary** or **Persistent** from the menu. The persistent option will maintain the frozen state after a reboot, given the user saves this change to the configuration.

### ▼ To freeze a service group from Command Center

1. Click **Commands>Operations>Availability>Freeze Service Group**.
2. Click the service group.
3. Click the **Persistent** check box if necessary. The persistent option maintains the frozen state after a reboot if you save this change to the configuration.
4. Click **Apply**.

## Unfreezing a Service Group

Thaw a frozen service group to perform online or offline operations on the service group.

### ▼ To thaw or “unfreeze” a service group from Cluster Explorer

1. In the **Service Groups** tab of the configuration tree, right-click the service group.  
*or*  
Click the cluster in the configuration tree, click the **Service Groups** tab, and right-click the service group icon in the view panel.
2. Click **Unfreeze**, and click **Temporary** or **Persistent** from the menu. The persistent option will maintain the thawed state after a reboot, given the user saves this change to the configuration.

### ▼ To thaw or “unfreeze” a service group from Command Center

1. Click Commands>Operations>Availability>Unfreeze Service Group.
2. Click the service group.
3. Click the **persistent** check box if necessary. The persistent option will maintain the thawed state after a reboot, given the user saves this change to the configuration.
4. Click **Apply**.



## Enabling a Service Group

Enable a service group before bringing it online. A service group that was manually disabled during a maintenance procedure on a system may need to be brought online after the procedure is completed.

### ▼ To enable a service group from Cluster Explorer

1. In the **Service Groups** tab of the configuration tree, right-click the service group.

*or*

Click the cluster in the configuration tree, click the **Service Groups** tab, and right-click the service group icon in the view panel.

2. Click **Enable**, and click the appropriate system from the menu.

### ▼ To enable a service group from Command Center

1. Click **Commands>Operations>Availability>Enable Service Group**.
2. Click the service group.
3. Click the system on which to enable the service group.
4. Click **Apply**.

## Disabling a Service Group

Disable a service group to prevent it from coming online. This process temporarily stops VCS from monitoring a service group on a system undergoing maintenance operations.

### ▼ To disable a service group from Cluster Explorer

1. In the **Service Groups** tab of the configuration tree, right-click the service group.  
*or*  
Click the cluster in the configuration tree, click the **Service Groups** tab, and right-click the service group icon in the view panel.
2. Click **Disable**, and click the appropriate system in the menu.

### ▼ To disable a service group from Command Center

1. Click **Commands>Operations>Availability>Disable Service Group**.
2. Click the service group.
3. Click the system on which to disable the service group.
4. Click **Apply**.



## Autoenabling a Service Group

A service group is aut disabled until VCS probes all resources and verifies they are ready to bring online. Autoenable a service group in situations where the VCS engine, HAD, is not running on one of the systems in the cluster, and you must override the disabled state of the service group to enable the group on another system.

### ▼ To autoenable a service group from Cluster Explorer

1. In the **Service Groups** tab of the configuration tree, right-click the service group.

*or*

Click the cluster in the configuration tree, click the **Service Groups** tab, and right-click the service group icon in the view panel.

2. Click **Autoenable**, and click the appropriate system from the menu.

### ▼ To autoenable a service group from Command Center

1. Click **Commands>Operations>Availability>Autoenable Service Group**.
2. Click the service group.
3. Click the system on which to autoenable the service group.
4. Click **Apply**.

## Flushing a Service Group

As a service group is brought online or taken offline, the resources within the group are also brought online and taken offline. If the online or offline operation is delayed on a particular resource, flush the service group to halt the operation on the resources waiting to go online or offline. Flushing a service group typically leaves the cluster in a partial state. After completing this process, resolve the issue with the particular resource (if necessary) and proceed with starting or stopping the service group.

### ▼ To flush a service group from Cluster Explorer

1. In the **Service Groups** tab of the configuration tree, right-click the service group.  
*or*  
Click the cluster in the configuration tree, click the **Service Groups** tab, and right-click the service group icon in the view panel.
2. Click **Flush**, and click the appropriate system from the menu.

### ▼ To flush a service group from Command Center

1. Click **Commands>Operations>Availability>Flush Service Group**.
2. Click the service group.
3. Click the system on which to flush the service group.
4. Click **Apply**.



## Linking Service Groups

### ▼ To link a service group from Cluster Explorer

1. Click a cluster in the configuration tree.
2. In the view panel, click the **Service Groups** tab. This opens the service group dependency graph.
3. To link a parent service group with a child service group, click the service group that will serve as the parent.
4. Drag the yellow line to the service group that will serve as the child.
5. Click the child group.
6. In the **Link Service Groups** dialog box, click the service group relationship and dependency type.



7. Click **OK**.

### ▼ To link a service group from Command Center

1. Click **Commands>Configuration>Dependencies>Link Service Groups**.
2. Click the parent resource group in the **Service Groups** box. After selecting the parent group, the potential groups that can serve as child groups are displayed in the **Child Service Groups** box.
3. Click a child service group.
4. Click the resource group relationship and dependency type.
5. Click **Apply**.

## Unlinking Service Groups

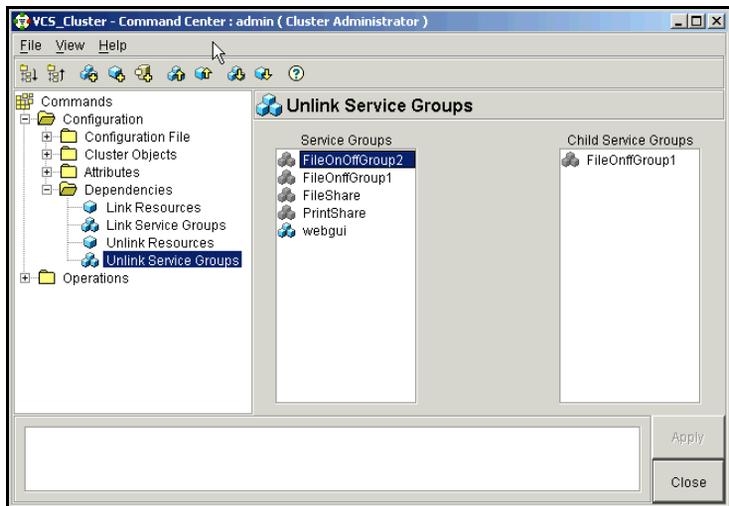
### ▼ To unlink a service group from Cluster Explorer

1. Click a cluster in the configuration tree.
2. In the view panel, click the **Service Groups** tab.
3. In the Service Group View, right-click the link between the service groups.
4. Click **Unlink** from the menu.
5. In the **Question** dialog box, click **Yes** to delete the link.



▼ To unlink a service group from Command Center

1. Click Commands>Configuration>Dependencies>Unlink Service Groups.
2. Click the parent service group in the **Service Groups** box. After selecting the parent group, the corresponding child groups are displayed in the **Child Service Groups** box.



3. Click the child service group.
4. Click **Apply**.

## Managing Systems for a Service Group

From Cluster Explorer, use the System Manager to add and remove systems in a service group's system list.

### ▼ To add a system to the service group's system list

1. In the **System Manager** dialog box, click the system in the **Available Systems** box.
2. Click the right arrow to move the available system to the **Systems for Service Group** table.
3. Click the check box in the **Startup** column to set the service group to start on the system automatically.
4. If necessary, double-click the entry in the **Priority** column and enter the new value. This number determines the sequence of systems that a service group fails over to if multiple systems exist.
5. Click **OK**.

### ▼ To remove a system from the service group's system list

1. In the **System Manager** dialog box, click the system in the **Systems for Service Group** table.
2. Click the left arrow to move the system to the **Available Systems** box.
3. Click **OK**.



## Creating Service Groups with the Configuration Wizard

Use the service group configuration wizard from Cluster Explorer or Configuration Editor to create a new service group. Configuration Editor enables you to complete this procedure on the local machine while VCS is offline.

### ▼ To create a service group

1. Open Configuration Wizard. (Click Tools>Configuration Wizard.)
2. Read the welcome screen and click **Next**.
3. Specify the name of the service group and the target systems on which it is to be configured.
  - a. Enter the name of the service group.
  - b. Click the target systems in the **Available Systems** box.
  - c. Click the right arrow to move the systems to the **Systems for Service Group** table. To remove a system from the table, click the system, then click the left arrow.
  - d. In the **Systems for Service Group** table, click the **Startup** check box to start the service group automatically on a system. The priority number indicates the failover sequence, with 1 denoting the first system to start after a failover.
4. Click **Next**.
5. Click **Next** again to configure the service group with a template and proceed to the next step. (Click **Finish** to add an empty service group to the selected cluster systems and configure it later.)
6. Click the template on which to base the new service group. The **Templates** box displays the templates available on the system to which Cluster Manager is connected. The resource dependency graph of the templates, the number of resources, and the resource types are also displayed.
7. Click **Next**. If a window notifies you that the name of a resource within the service group is already in use, proceed to the next step. Otherwise, proceed to [step 9](#).

8. Click **Next** to automatically apply all of the new names listed in the table to resolve the name clash.  
*or*  
Modify the clashing names by entering text in the field next to the **Apply** button, selecting the location of the text for each name from the Correction drop-down list box, clicking **Apply**, and clicking **Next**.
9. Click **Next** to create the service group. A window displays the commands that are adding the group, its resources, and the attributes and dependencies specified in the template. A progress indicator displays the percentage of the commands fired. The actual commands are displayed at the top of the indicator.
10. After the service group is successfully created, click **Next** to edit attributes using the wizard and proceed to the next step. (Click **Finish** to edit attributes later using Cluster Explorer.)
11. Review the attributes associated with the resources of the service group. If necessary, proceed to the next step to modify the default values of the attributes. Otherwise, proceed to [step 13](#) to accept the default values and complete the configuration.
12. Modify the values of the attributes (if necessary).
  - a. Click the resource.
  - b. Click the attribute to be modified.
  - c. Click the **Edit** icon at the end of the table row.
  - d. In the **Edit Attribute** dialog box, enter the attribute values.
  - e. Click **OK**.
  - f. Repeat the procedure for each resource and attribute.
13. Click **Finish**.



## Administering Resources

The Java Console enables you to administer resources in the cluster. Use it to add and delete, bring online and take offline, probe, enable and disable, clear, and link and unlink resources. You can also import resource types to the configuration.

### Defining Critical and Enabled Resource Attributes

Use the Java Console to click Critical and Enabled attributes for resources when adding them to a service group.

- ◆ A critical resource indicates that the service group is faulted when the resource, or any resource it depends on, faults. The default value for the Critical attribute is 1.
- ◆ An enabled resource indicates agents monitor the resource. If a resource is created dynamically while VCS is running, you must enable the resource before VCS monitors it. When Enabled is set to 0, it implies a disabled resource. VCS will not bring a disabled resource nor its children online, even if the children are enabled. If you specify the resource in the main.cf prior to starting VCS, the default value for the Enabled attribute is 1.

### Adding a Resource

The Java Console provides several ways to add a resource to a service group. Use the Cluster Explorer, Command Center, Template View, or Configuration Editor to perform this task.

#### ▼ To add a resource from Cluster Explorer

1. On the **Edit** menu, click **Add**, and click **Resource**.

*or*

In the **Service Group** tab of the configuration tree, right-click the service group or the resource type and click **Add Resource** from the menu.

*or*

Click **Add Resource** in the Cluster Explorer toolbar.

2. In the **Add Resource** dialog box, enter the name of the resource.

**Add Resource**

Group name : FILEGROUP1

Resource name

Resource Type

Attribute na...	Type	Dimension	Value	Edit
AgentDebug	Boolean	Scalar	false	
PathName	String	Scalar		

Critical  Enabled

3. Click the resource type from the drop-down list box.
4. Edit resource attributes according to your configuration. (You can also edit attributes after adding the resource.)
5. Click the **Critical** and **Enabled** check boxes, if applicable.
6. Click **Show Command** in the bottom left corner if you want to view the command associated with the resource. Click **Hide Command** to close the view of the command
7. Click **OK**.

#### ▼ To add a resource from Command Center

1. Click **Commands>Configuration>Cluster Objects>Add Resource**.  
*or*  
Click **Add resource** in the Command Center toolbar.
2. Enter the name of the resource.
3. Click the resource type from the **Resource Type** box.
4. Edit resource attributes according to your configuration. (You can also edit attributes after adding the resource.)



5. Click the **Critical** and **Enabled** check boxes, if applicable.
6. Click **Apply**.

▼ **To add a resource from the Template View**

1. On the Cluster Explorer **Tools** menu, click **Templates**.

*or*

On the Configuration Editor **Tools** menu, click **Templates**.

2. In the left pane of the Template View, click the template from which to add resources to your configuration.
3. In the resource graph, right-click the resource to be added to your configuration. Click **Copy**, and click **Self** from the menu to copy the resource.

*or*

In the resource graph, right-click the resource to be added to your configuration. Click **Copy**, and click **Self and Child Nodes** from the menu to copy the resource with its dependent resources.

4. In the **Service Groups** tab of the Cluster Explorer configuration tree, click the service group to which to add the resources.
5. In the Cluster Explorer view panel, click the **Resources** tab.
6. Right-click the Resource View panel and click **Paste** from the menu. After the resources are added to the service group, edit the attributes to configure the resources.

### ▼ To add a resource from Configuration Editor

Use Configuration Editor to add a resource on the local system while VCS is offline. Some options in the following procedure may not be available, depending on the specifications in the configuration file.

1. Click the service group in the configuration tree. On the **Edit** menu, click **Add**, and click **Resource**.

*or*

In the configuration tree, right-click the service group and click **Add Resource** from the menu.

*or*

Click the service group in the configuration tree and click **Add Resource** in the Configuration Editor toolbar.

2. In the **Add Resource** dialog box, enter the name of the resource.
3. Click the resource type.
4. Edit resource attributes according to your configuration. (You can also edit attributes after adding the resource.)
5. Click the **Critical** and **Enabled** check boxes, if applicable.
6. Click **Show Command** in the bottom left corner if you want to view the command associated with the resource. Click **Hide Command** to close the view of the command.
7. Click **OK**.



## Deleting a Resource

### ▼ To delete a resource from Cluster Explorer

1. In the **Service Groups** tab of the configuration tree, right-click the resource.

*or*

Click a service group in the configuration tree, click the **Resources** tab, and right-click the resource icon in the view panel.

2. Click **Delete** from the menu.

### ▼ To delete a resource from Command Center

1. Click **Commands>Configuration>Cluster Objects>Delete Resource**.

2. Click the resource.

3. Click **Apply**.

### ▼ To delete a resource from Configuration Editor

1. Click the resource in the configuration tree.

2. On the **Edit** menu, click **Delete**, and click **Resource**.

*or*

Right-click the resource in the configuration tree, and click **Delete** from the menu.

## Bringing a Resource Online

### ▼ To bring a resource online from Cluster Explorer

1. In the **Service Groups** tab of the configuration tree, right-click the resource.

*or*

Click a service group in the configuration tree, click the **Resources** tab, and right-click the resource icon in the view panel.

2. Click **Online**, and click the appropriate system from the menu.

### ▼ To bring a resource online from Command Center

1. Click **Commands>Operations>Controls>Online Resource**.



2. Click a resource.
3. Click a system on which to bring the resource online.
4. Click **Apply**.



## Taking a Resource Offline

### ▼ To take a resource offline from Cluster Explorer

1. In the **Service Groups** tab of the configuration tree, right-click the resource.

*or*

Click a service group in the configuration tree, click the **Resources** tab, and right-click the resource icon in the view panel.

2. Click **Offline**, and click the appropriate system from the menu.

### ▼ To take a resource offline from Command Center

1. Click **Commands>Operations>Controls>Offline Resource**.
2. Click a resource.
3. Click a system on which to take the resource offline.
4. Click **Apply**.



## Taking a Resource and all Child Resources Offline

The Java Console enables you to offline a parent resource and all of its child resources. Use the OffProp Resource feature to propagate the offline state of the parent resource.

### ▼ To take a parent resource and its child resources offline from Cluster Explorer

1. In the **Service Groups** tab of the configuration tree, right-click the resource.
2. Click **Offline Prop**, and click the appropriate system from the menu.

### ▼ To take a parent resource and its child resources offline from Command Center

1. Click **Commands>Operations>Controls>OffProp Resource**.
2. Click the resource.
3. Click the system on which to offline the resource and its child resources.
4. Click **Apply**.

## Probing a Resource

Probe a resource to check that it is configured and ready to bring online.

### ▼ To probe a resource from Cluster Explorer

1. In the **Service Groups** tab of the configuration tree, right-click the resource.
2. Click **Probe** and then click the appropriate system from the menu.

### ▼ To probe a resource from Command Center

1. Click **Commands>Operations>Controls>Probe Resource**.
2. Click the resource.
3. Click the system on which to probe the resource.
4. Click **Apply**.



## Enabling Resources in a Service Group

Enable resources in a service group to bring the disabled resources online. A resource may have been manually disabled to temporarily stop VCS from monitoring the resource.

### ▼ To enable an individual resource in a service group

1. From **Cluster Explorer**, click the Service Groups tab in the Cluster Explorer configuration tree.
2. Right-click a resource in the configuration tree, and click **Enabled** from the menu.

### ▼ To enable all resources in a service group

1. From **Cluster Explorer**, click the **Service Groups** tab in the Cluster Explorer configuration tree, right-click the service group, and click **Enable Resources**.

*or*

From **Command Center**, go to Commands>Operations>Availability>Enable Resources for Service Group and proceed to the next step.

2. Click the service group.
3. Click **Apply**.

## Disabling Resources in a Service Group

Disable resources in a service group to prevent them from being brought online. This disabling process is useful when you want VCS to temporarily “ignore” resources (rather than delete them) while the service group is online.

### ▼ To disable an individual resource in a service group

1. From **Cluster Explorer**, click the **Service Groups** tab in the Cluster Explorer configuration tree.
2. Right-click a resource in the configuration tree. An enabled resource will display a check mark next to the **Enabled** option that appears in the menu.
3. Click **Enabled** from the menu to clear this option.

### ▼ To disable all resources in a service group

1. From **Cluster Explorer**, click the Service Groups tab in the Cluster Explorer configuration tree, right-click the service group, and click **Disable Resources**.

*or*

From **Command Center**, click Commands>Operations>Availability>Disable Resources for Service Group and proceed to the next step.

2. Click the service group.
3. Click **Apply**.

## Clearing a Resource

Clear a resource to remove a fault and make the resource available to go online. A resource fault can occur in a variety of situations, such as a power failure or a faulty configuration.

### ▼ To clear a resource from Cluster Explorer

1. In the **Service Groups** tab of the configuration tree, right-click the resource.
2. Click **Clear** and then click the appropriate system from the menu.

### ▼ To clear a resource from Command Center

1. From **Command Center**, click Commands>Operations>Availability>Clear Resource.
2. Click the resource.
3. Click the system on which to clear the resource.
4. Click **Apply**.



## Linking Resources

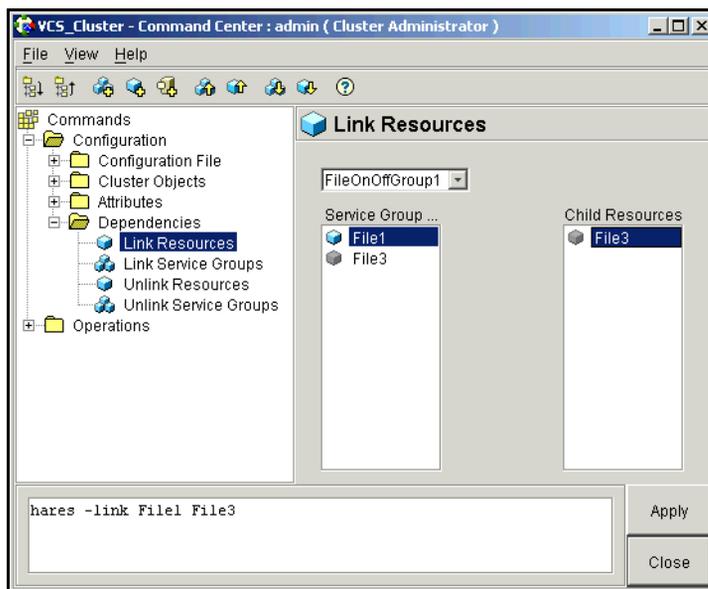
Use Cluster Explorer, Command Center, or Configuration Editor to link resources in a service group. Configuration Editor enables you to link resources on the local system while VCS is offline.

### ▼ To link resources from Cluster Explorer

1. In the configuration tree, click the **Service Groups** tab.
2. Click the service group to which the resources belong.
3. In the view panel, click the **Resources** tab. This opens the resource dependency graph.
4. To link a parent resource with a child resource, click the resource that will serve as the parent.
5. Drag the yellow line to the resource that will serve as the child.
6. Click the child resource.
7. In the **Confirmation** dialog box, click **Yes**.

## ▼ To link resources from Command Center

1. Click Commands>Configuration>Dependencies>Link Resources.



2. In the drop-down list box, click the service group whose resources are to be linked.
3. Click the parent resource in the **Service Group Resources** box. After selecting the parent resource, the potential resources that can serve as child resources are displayed in the **Child Resources** box.
4. Click a child resource.
5. Click **Apply**.



▼ **To link resources from Configuration Editor**

1. In the configuration tree, click the service group to which the resources belong. This opens the resource dependency graph.
2. To link a parent resource with a child resource, click the resource that will serve as the parent.
3. Drag the yellow line to the resource that will serve as the child.
4. Click the child resource.
5. In the **Confirmation** dialog box, click **Yes**.

## Unlinking Resources

Use Cluster Explorer, Command Center, or Configuration Editor to unlink resources in a service group. Configuration Editor enables you to unlink resources on the local system while VCS is offline.

▼ **To unlink resources from Cluster Explorer**

1. In the configuration tree, click the **Service Groups** tab.
2. Click the service group to which the resources belong.
3. In the view panel, click the **Resources** tab.
4. In the Resource View, right-click the link between the resources.
5. Click **Unlink** from the menu.
6. In the **Question** dialog box, click **Yes** to delete the link.

---

### ▼ To unlink resources from Command Center

1. Click **Commands>Configuration>Dependencies>Unlink Resources**.
2. In the drop-down list box, click the service group whose resources are to be unlinked.
3. Click the parent resource in the **Service Group Resources** box. After selecting the parent resource, the corresponding child resources are displayed in the **Child Resources** box.
4. Click the child resource.
5. Click **Apply**.

### ▼ To unlink resources from Configuration Editor

1. In the configuration tree, click the service group to which the resources belong. This opens the resource dependency graph.
2. Right-click the link between the resources.
3. Click **Unlink** from the menu.
4. In the **Question** dialog box, click **Yes** to delete the link.



## Importing Resource Types

The Java Console enables you import resource types to your configuration (main.cf). For example, use this procedure to import the types.cf for enterprise agents to your configuration (if necessary). You cannot import resource types that already exist in your configuration.

### ▼ To import a resource type from Cluster Explorer

1. On the **File** menu, click **Import Types**.
2. In the **Import Type** dialog box, click the file from which to import the resource type. The dialog box displays the files on the system that Cluster Manager is connected to.
3. Click **Import**.

### ▼ To import a resource type from Configuration Editor

1. On the **File** menu, click **Import Types**.
2. Click the file from which to import the resource type. The dialog box displays the files on the system that Cluster Manager is connected to.
3. Click **Open**.

## Administering Systems

The Java Console enables you to administer systems in the cluster. Use it to add, delete, freeze, and unfreeze systems. (Configuration Editor enables you to administer systems on the local system while VCS is offline.)

### Adding a System

Cluster Explorer, Command Center, and Configuration Editor enable you to add a system to the cluster. A system must have an entry in the `/etc/llttab` configuration file before it can be added to the cluster.

#### ▼ To add a system from Cluster Explorer

1. On the **Edit** menu, click **Add**, and click **System**.

*or*

In the **System** tab of the Cluster Explorer configuration tree, right-click the cluster and click **Add System** from the menu.

*or*

Click **Add System** in the Cluster Explorer toolbar.

2. Enter the name of the system.
3. Click **Show Command** in the bottom left corner if you want to view the command associated with the system. Click **Hide Command** to close the view of the command.
4. Click **OK**.

#### ▼ To add a system from Command Center

1. Click **Commands>Configuration>Cluster Objects>Add System**.

*or*

Click **Add system** in the Command Center toolbar.

2. Enter the name of the system.
3. Click **Apply**.



▼ **To add a system from Configuration Editor**

1. Click the Systems folder in the configuration tree. On the **Edit** menu, click **Add**, and click **System**.

*or*

Right-click the Systems folder, click **Add**, and click **System** from the menu.

*or*

Click **Add System** in the Configuration Editor toolbar.

2. Enter the name of the system.
3. Click **Show Command** in the bottom left corner if you want to view the command associated with the system. Click **Hide Command** to close the view of the command.
4. Click **OK**.

## Deleting a System

Use Command Center or Configuration Editor to delete a system.

▼ **To delete a system from Command Center**

1. Click Commands>Configuration>Cluster Objects>Delete System.
2. Click the system.
3. Click **Apply**.

▼ **To delete a system from Configuration Editor**

Click the system in the configuration tree. On the **Edit** menu, click **Delete**, and click **System**.

*or*

Right-click the system in the configuration tree and click **Delete** from the menu.

## Freezing a System

Freeze a system to prevent its components from failing over to another system. Use this procedure during a system upgrade.

### ▼ To freeze a system from Cluster Explorer

1. Click the **Systems** tab of the configuration tree.
2. In the configuration tree, right-click the system, click **Freeze**, and click **Temporary** or **Persistent** from the menu. The persistent option maintains the frozen state after a reboot if the user saves the change to the configuration.

### ▼ To freeze a system from Command Center

1. Click Commands>Operations>Availability>Freeze System.
2. Click the system.
3. Click the **evacuate** check box or the **persistent** check box (if necessary). The evacuate option moves all service groups to a different system before the freeze operation occurs. The persistent option maintains the frozen state after a reboot if the user saves the change to the configuration.
4. Click **Apply**.



## Unfreezing a System

Thaw a frozen system to perform online and offline operations on the system.

### ▼ To thaw or “unfreeze” a system from Cluster Explorer

1. Click the **Systems** tab of the configuration tree.
2. In the configuration tree, right-click the system, click **Unfreeze**, and click **Temporary** or **Persistent** from the menu. The persistent option maintains the thawed state after a reboot if the user saves the change to the configuration.

### ▼ To thaw or “unfreeze” a system from Command Center

1. Click **Commands>Operations>Availability>Unfreeze System**.
2. Click the system.
3. Click the **persistent** check box (if necessary). The persistent option maintains the thawed state after a reboot if the user saves the change to the configuration.
4. Click **Apply**.



## Administering the Cluster Configuration

The Java Console enables you to modify the VCS configuration. The configuration defines the parameters of the entire cluster.

- ◆ Use Cluster Explorer, Command Center, or Configuration Editor to open and save a configuration. Configuration Editor enables you to administer the configuration on the local system while VCS is offline.
- ◆ Use Cluster Explorer or Command Center to close a configuration.

### Opening a Cluster Configuration

Modify a read-only configuration file to a read/write file by opening the configuration from Cluster Explorer, Command Center, or Configuration Editor.

#### ▼ To open a configuration from Cluster Explorer

On the **File** menu, click **Open Configuration**.

*or*

Click **Open Configuration** in the Cluster Explorer toolbar.

*or*

Right-click the cluster in the configuration tree, and click **Open Configuration** from the menu.

#### ▼ To open a configuration from Command Center

1. Click **Commands>Configuration>Configuration File>Open Configuration**.
2. Click **Apply**.

#### ▼ To open a configuration from Configuration Editor

Click **Open Configuration** in the Configuration Editor toolbar.



## Saving a Cluster Configuration

After updating the VCS configuration, use Cluster Explorer, Command Center, or Configuration Editor to save your latest configuration to disk as a read/write file.

### ▼ To save a configuration from Cluster Explorer

On the **File** menu, click **Save Configuration**.

*or*

Click **Save Configuration** in the Cluster Explorer toolbar.

*or*

Right-click the cluster in the configuration tree, and click **Save Configuration** from the menu.

### ▼ To save a configuration from Command Center

1. Click **Commands>Configuration>Configuration File>Save Configuration**.
2. Click **Apply**.

### ▼ To save a configuration from Configuration Editor

On the **File** menu, click **Save**.

*or*

Click **Save Configuration** in the Configuration Editor toolbar.

## Closing a Cluster Configuration

After updating the VCS configuration, use Cluster Explorer or Command Center to save your latest configuration to disk as a read-only file.

### ▼ To close a configuration from Cluster Explorer

On the **File** menu, click **Close Configuration**.

*or*

Click **Close Configuration** in the Cluster Explorer toolbar.

*or*

Right-click the cluster in the configuration tree, and click **Close Configuration** from the menu.

### ▼ To close a configuration from Command Center

1. Click **Commands>Configuration>Configuration File>Close Configuration**.
2. Click **Apply**.



## Using Configuration Editor

The Configuration Editor window enables you to build new configuration files or modify existing configuration files on the local system. If you modify a configuration, load the existing `main.cf` or `types.cf` file, modify it using the options available in the Configuration Editor interface, and save the configuration on the local system. To start VCS with a modified configuration, save the configuration file in the directory `$VCS_HOME/conf/config` on the system.

Use the Offline Configuration Wizard in Configuration Editor to build or modify a configuration not currently connected to a particular host. (Use the Configuration Wizard in Cluster Explorer to modify a configuration while connected to a specific host.)

### Building a New Configuration in Configuration Editor

1. Start Configuration Editor ([page 118](#)).
2. On the **Offline Configuration Wizard** dialog box that appears after launching the **Configuration Editor**, click **New Configuration**.
3. Click **Next**.
4. Enter the path name of the `types.cf` file that will serve as the basis of the configuration.
5. Click **Finish**.
6. When prompted to use a wizard to add service groups, click **Yes** and proceed to the next step. (Click **No** to add service groups later.)
7. Return to [step 2](#) of “[Creating Service Groups with the Configuration Wizard](#)” on [page 172](#).

## Modifying a Configuration in Configuration Editor

1. On the **Offline Configuration Wizard** dialog box that appears after you launch Configuration Editor, click **Open Configuration**.
2. Click **Next**.
3. Enter the path name of the configuration folder.
4. Click **Finish**.
5. When prompted to use a wizard to add service groups, click **Yes** and proceed to the next step. (Click **No** to add service groups later.)
6. Return to [step 2](#) of “[Creating Service Groups with the Configuration Wizard](#)” on page 172.



## Running Commands

Use Command Center and Cluster Shell to run commands on a cluster. Command Center enables you to run commands organized as “Configuration” and “Operation.” Cluster Shell enables you to run a non-interactive shell command on one or more cluster systems.

### ▼ To run a command from Command Center

1. From **Command Center**, click the command from the command tree. If necessary, expand the tree to view the command.
2. In the corresponding command interface, click the cluster objects and appropriate options (if necessary).
3. Click **Apply**.

### ▼ To run a non-interactive shell command

1. From **Cluster Shell**, click the system in the **Target Systems** box. To click all systems, click **Select all**. To clear all of the systems, click **Deselect all**.
2. Enter the command in the bottom pane.
3. Click **Send**.

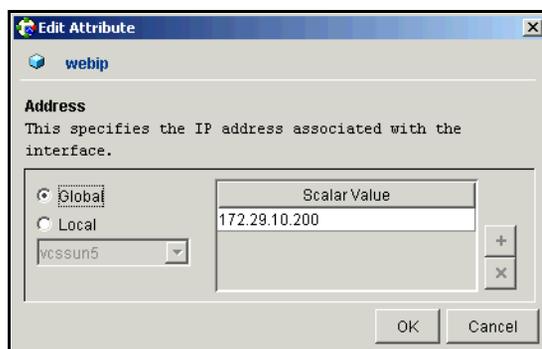


## Editing Attributes

The Java Console enables you to edit attributes of cluster objects. For a complete list of VCS attributes, see [Appendix B](#). By default, the Java Console displays key attributes and type-specific attributes. To view all attributes associated with a cluster object, click **Show all attributes**.

### ▼ To edit an attribute from Cluster Explorer

1. In the Cluster Explorer configuration tree, click the object whose attributes you want to edit.
2. In the view panel, click the **Properties** tab. If the attribute does not appear in the Properties View, click **Show all attributes**. This opens the Attributes View.
3. In the Properties View, click the icon in the Edit column of the **Key Attributes** or **Type Specific Attributes** table. In the Attributes View, click the **Edit icon** in the Edit column of the attribute table.
4. In the **Edit Attribute** dialog box, enter changes to the attributes values and click **OK**.



- *To edit a scalar value:*  
Enter or click the value.
- *To edit a non-scalar value:*  
Use the + button to add an element. Use the x button to delete an element.
- *To change the attribute's scope:*  
Click the **Global** or **Local** option.
- *To change the system for a local attribute:*  
Click the system from the drop-down list box.



▼ **To edit an attribute from Command Center**

1. Click **Commands>Configuration>Attributes>Modify *cluster\_object* Attributes**.
2. Click the cluster object from the drop-down list box.
3. In the attribute table, click the **Edit icon** in the Edit column of the attribute.
4. In the **Edit Attributes** dialog box, enter the changes to the attributes values.
  - *To edit a scalar value:*  
Enter or click the value.
  - *To edit a non-scalar value:*  
Use the + button to add an element. Use the x button to delete an element.
  - *To change the attribute's scope:*  
Click the **Global** or **Local** option.
  - *To change the system for a local attribute:*  
Click the system from the drop-down list box.
5. Click **OK**.

▼ **To edit an attribute from Configuration Editor**

1. Click the object in the configuration tree. On the **Edit** menu, click **Modify Attributes**.  
*or*  
Right-click the object in the configuration tree and then click **Modify Attributes** from the menu.
2. In the attribute table, click the **Edit icon** in the Edit column of the attribute.
3. In the **Edit Attributes** dialog box, enter the changes to the attributes values.
  - *To edit a scalar value:*  
Enter or click the value.
  - *To edit a non-scalar value:*  
Use the + button to add an element. Use the x button to delete an element.



- *To change the attribute's scope:*  
Click the **Global** or **Local** option.
  - *To change the system for a local attribute:*  
Click the system from the drop-down list box.
4. Click **OK**.

## Customizing the Log Display

The Java Console enables you to customize the log display of messages generated by the VCS engine, HAD. On the Log Desk, you can set filter criteria to search and view messages.

To browse the Log Desk for detailed views of each log message or command, double-click the event's description. Use the arrows in the Event Log pop-up window to navigate backward and forward through the message list.

### ▼ To customize the event log display

1. In the **Logs of** drop-down list box, click the category of log messages to display.
2. In the **Named** drop-down list box, click the name of the cluster object whose log messages you want to display. To view all the messages for the selected category, click **All**.
3. In the **Logs from last** field, enter the numerical value and click the time unit from the drop-down list box.



## Uninstalling Cluster Manager (Java Console)

1. Open the Control Panel. (Click Start>Settings>Control Panel.)
2. Double-click **Add/Remove Programs**.
3. On the **Add/Remove Programs** window, click **VERITAS Cluster Manager (Java Console)**.
4. Click **Remove**.
5. Click **Yes** in the confirmation dialog box. A Setup window displays the status of the removal.



# Administering VCS from Cluster Manager (Web Console)

---

8

## About the Web Console

This Web-based graphical user interface enables you to monitor the cluster and perform basic cluster administration, including many of the same operations as the command-line interface and Java Console. From the Web Console you can:

- ◆ Connect to a remote cluster using a Web browser and view the status of the cluster and its components in HTML “pages.”
- ◆ Open and close cluster configurations.
- ◆ Bring service groups and resources online and take them offline.
- ◆ Switch, enable, disable, freeze, unfreeze, autoenable, and flush service groups.
- ◆ Clear and probe resources.
- ◆ Freeze and unfreeze systems.

---

**Note** You cannot log on to the Web Console unless you have enabled Java, JavaScript, and cookies on the browser (Internet Explorer or Netscape Navigator, version 4.0 or later).

---



## GUI Conventions

The Web Console uses the following icons to represent VCS components.

Icon	VCS Component
	Service Group
	System
	Resource
	Resource Type

- ◆ Color icons indicate components are in a healthy running condition.
- ◆ Gray icons indicate components are offline.
- ◆ Icons with orange lines indicate components partially online with only some resources online.
- ◆ Icons with red lines indicate faulted components.
- ◆ Icons with a question mark indicate components whose status is unknown.

## Web Console Views

There are 10 HTML “pages” in the Web Console. Each page displays information about the cluster or its objects.

**Note** The examples in this chapter are generic. Actual page views, including service groups, attributes and resource types, will vary for each operating system.

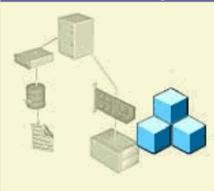
### Cluster Summary Page

The Cluster Summary page is displayed when you log on to the Web Console. It includes summary information about the service groups, systems, and logs.

The Service Groups summary in the left pane displays the status of the service groups; for example, how many service groups are online and how many are offline. The Systems summary in the right pane displays the status of each system in the cluster. The Logs summary in the bottom pane displays the five most recent critical/error messages.

**Summary information for cluster - VCScluster3738**

**Service Groups**



Online : 1 Group

**Systems**



All Systems are Running

**Recent Critical/Error Messages** Current Time at Server : Aug 10, 2001 4:03:19 PM

Severity	Log Message	Time Stamp
Error	VCS:10322:System VCSW2K38 (Node '1') changed state from RUNNING to FAULTED	8/10/01 4:57 AM
Error	(VCSW2K37) VCS:13005:Agent(Process) is exiting due to internal error.	8/9/01 4:29 PM
Error	VCS:10322:System VCSW2K38 (Node '1') changed state from RUNNING to FAULTED	8/9/01 4:28 PM
Error	VCS:10205:Group ClusterService is faulted on system VCSW2K38	8/9/01 4:26 PM
Error	VCS:10322:System VCSW2K37 (Node '0') changed state from RUNNING to FAULTED	8/9/01 4:22 PM

## All Groups Page

The All Groups page displays summary status of service groups throughout the cluster. Click a service group name in the left column for details about the group.

- ✓ To access **All Groups**, click **Groups** in the top right corner of any page.

All Groups			
	Group Name	Status	Other Information
	<a href="#">FileOnOffGroup2</a>	Online on VCSNT28	Disabled , Frozen
	<a href="#">FileShare</a>	Offline on All Systems	-
	<a href="#">FileShareGroup2</a>	Offline on All Systems	Auto Disabled
	<a href="#">PrintShare</a>	Offline on All Systems	-
	<a href="#">webgui</a>	Online on VCSNT25	-

## All Systems Page

The All Systems page displays the status of systems throughout the cluster and lists the ONLINE and FAULTED service groups on the systems. Click a service group or system name for details about the group or system.

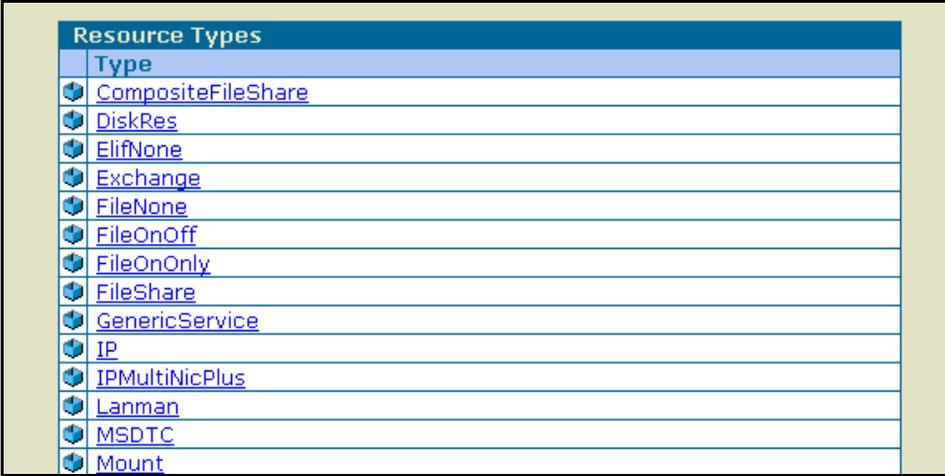
- ✓ To access **All Systems**, click **Systems** in the top right corner of any page.

All Systems				
	System Name	System State	Online Groups	Faulted Groups
	<a href="#">VCSNT25</a>	Running	<a href="#">webgui</a>	
	<a href="#">VCSNT26</a>	Running		
	<a href="#">VCSNT27</a>	Exited		
	<a href="#">VCSNT28</a>	Running	<a href="#">FileOnOffGroup2</a>	

## All Types Page

The All Types page displays resource types that can be configured in your cluster. Click the name of a resource type for details about the type.

- ✓ To access **All Types**, click **Types** in the top right corner of any page.



Resource Types	
Type	
 <a href="#">CompositeFileShare</a>	
 <a href="#">DiskRes</a>	
 <a href="#">ElifNone</a>	
 <a href="#">Exchange</a>	
 <a href="#">FileNone</a>	
 <a href="#">FileOnOff</a>	
 <a href="#">FileOnOnly</a>	
 <a href="#">FileShare</a>	
 <a href="#">GenericService</a>	
 <a href="#">IP</a>	
 <a href="#">IPMultiNicPlus</a>	
 <a href="#">Lanman</a>	
 <a href="#">MSDTC</a>	
 <a href="#">Mount</a>	



## All Attributes Page

The All Attributes page lists the attributes associated with the cluster and its components. Each attribute includes its scope, dimension, and value. For example, the value of a service group's SystemList attribute specifies the system on which the group is configured and the priority of each system within the group. The page also enables you to edit some attributes.

- ✓ To access this page, click **All Attributes** from the **Cluster Summary** page.

All Attributes					
	Name	Scope	Dimension	Values	Edit
	Administrators	Global	Keylist	admin	
	AllowNativeCliUsers	Global	Scalar	false	
	ClusterLocation	Global	Scalar		
	ClusterName	Global	Scalar	VCSCluster3738	
	ClusterOwner	Global	Scalar		
	CompareRSM	Global	Scalar	0	
	CounterInterval	Global	Scalar	5	
	DumpingMembership	Global	Scalar	0	
	EngineClass	Global	Scalar	RT	
	EnginePriority	Global	Scalar		
	GlobalCounter	Global	Scalar	0	
	GroupLimit	Global	Scalar	200	
	HacliUserLevel	Global	Scalar	NONE	
	LockMemory	Global	Scalar	ALL	
	LogSize	Global	Scalar	33554432	
	MajorVersion	Global	Scalar	2	
	MinorVersion	Global	Scalar	0	

## Log Details Page

The Log Details page displays log messages generated by the VCS engine, HAD. By default, each log view displays 10 messages. The icon in the first column of the table indicates the severity level of the message. You can customize this view by setting filter criteria. See “[Customizing the Log Display](#)” on page 233 for more information. Click the page numbers or the right/left arrows to navigate through the messages.

- ✓ To access **Log Details**, click **Logs** in the top right corner of any page.

Current Time at Server : Aug 10, 2001 4:31:06 PM		
<span>✖ Critical</span> <span>⊖ Error</span> <span>⚠ Warning</span> <span>⚡ Action</span> <span>ℹ Normal</span>		
<b>Log Messages</b> <span>◀</span> <span>Pages :</span> <span>1</span> <span>2</span> <span>3</span> <span>4</span> <span>5</span> <span>6</span> <span>7</span> <span>8</span> <span>9</span> <span>10</span> <span>20</span> <span>30</span> <span>40</span> <span>50</span> <span>▶</span>		
	Log Messages	Time Stamp
⚠	VCS:10526:IpmHandle::recv peer exited errno 10054	2001/08/10 16:09:04
ℹ	VCS:50106:User Administrator fired command: haconf -dump -makero	2001/08/10 16:09:03
ℹ	VCS:50106:User Administrator fired command: hatype -delete IIS	2001/08/10 16:09:03
ℹ	VCS:50106:User Administrator fired command: haconf -makerw	2001/08/10 16:09:02
⚠	VCS:10526:IpmHandle::recv peer exited errno 10054	2001/08/10 16:08:10
ℹ	VCS:50106:User Administrator fired command: haconf -dump -makero	2001/08/10 16:08:09
ℹ	VCS:50106:User Administrator fired command: hatype -modify IIS LogFileSize 65535	2001/08/10 16:08:09
ℹ	VCS:50106:User Administrator fired command: hatype -add IIS	2001/08/10 16:08:09
ℹ	VCS:50106:User Administrator fired command: haconf -makerw	2001/08/10 16:08:09
ℹ	VCS:10056:User admin has logged in	2001/08/10 16:03:25



## Service Group Page

The Service Group page displays the group’s status and key attributes, and a list of configured resources and their status. For the entire selection of attributes associated with the service group, click All Attributes in top right corner of the Attributes table.

- ✓ To access this page, click the name of a service group from the **All Groups** page.

The screenshot shows the 'ClusterService' page in the VCS Cluster Web Console. At the top, there are navigation links: [Online](#), [Offline](#), [Switch](#), [Enable](#), [Disable](#), [Freeze](#), [Unfreeze](#), [Autoenable](#), and [Flush](#). Below these, the group name is 'ClusterService' and the status is 'Online on VCSW2K81'.

The 'Attributes' table is as follows:

Attributes		All Attributes
Name	Value	Edit
Parallel	false	...
Failover Policy	Priority	...
Auto Failover	true	...
Auto Restart	true	...
Frozen	false	...
Frozen temporarily	false	...

The 'Status Information on Member Systems' table is as follows:

System Name	State	Auto Disabled	Enabled
VCSW2K81	Online	False	True
VCSW2K82	Offline	False	True

The 'Resource List' table is as follows:

Name	Type	Status
VRTSweb	Process	Online
webip	IP	Online

At the bottom, there is a section for 'Recent Logs for this Object' with a sub-header 'Current Time at Server : Dec 28, 2001 4:33:53 PM' and a table with columns 'Severity/Log Message' and 'Time Stamp'.



## System Page

The System page displays the system state, key attributes, and details of service groups configured on the system. For the entire selection of attributes associated with the system, click All Attributes in top right corner of the Attributes table.

- ✓ To access this page, click the name of a system from the **All Systems** page. (Click Cluster Summary>All Systems>*system*.)

VCSNT25-28 → All Systems → VCSW2K82

System Name : VCSW2K82  
System State : Running

Freeze Unfreeze

VCSW2K82		All Attributes
Attributes	Value	Edit
Frozen	false	...
Frozen temporarily	false	...
Online Group Count	0	
System State	Running	
Up/Down State	Up	

Details of Groups Configured on this system

Group Name	Status
<a href="#">ClusterService</a>	Online on VCSW2K81
<a href="#">connect</a>	Faulted on VCSW2K81, VCSW2K82

Recent Logs for this Object Current Time at Server : Dec 28, 2001 4:30:08 PM

Severity	Log Message	Time Stamp
⚠	VCS:10322: System VCSW2K82 (Node '1') changed state from LOCAL_BUILT to RUNNING	12/28/01 4:01 PM



## Resource Type Page

The Resource Type page displays the resource type and its key attributes. For the entire selection of attributes associated with the resource type, click **All Attributes** in top right corner of the Attributes table. To edit attributes, click the Edit icon (...).

- ✓ To access this page, click the name of a resource type from the **All Types** page. (Click Cluster Summary>All Types>*resource type*.)

DiskRes		All Attributes
Attributes	Value	Edit
ArgList	AgentDebug Signatures ConfigPercentage FailFastInterval GracePeriod	...
Operations	OnOff	...
Restart Limit	0	...
Online Timeout	300	...
Offline Timeout	300	...
Monitor Timeout	60	...
Monitor Interval	60	...
LogLevel	error	...

## Resource Page

The Resource page displays the resource status and key attributes. For the entire selection of attributes associated with the resource, click All Attributes in top right corner of the Attributes table. For more information on operations available from this page, see “[Administering Resources](#)” on page 226.

- ✓ To access this page, click a resource name from the **Service Group** page (Cluster Summary>All Groups>*service group*).

The screenshot shows the Web Console interface for the resource 'webip'. The breadcrumb navigation is 'VCSCluster > All Groups > ClusterService > webip'. The resource status is 'Online on VCSW2K81'. The main content area contains a table of attributes for 'webip', a table of resource details per system, and a section for recent logs.

Resource Name : webip  
 Type : IP  
 Status : Online on VCSW2K81

webip		All Attributes
Attributes	Value	Edit
Critical	true	...
Enabled	true	...
Address	10.212.96.205	...
SubNetMask	255.255.254.0	...
MACAddress	VCSW2K81 - 00-B0-D0-79-0F-96 VCSW2K82 - 00-B0-D0-79-0F-98	...
Agent Debug	false	...

Resource Details per System				
System Name	State	IState	Confidence Level	Flags
VCSW2K81	Online	Not Waiting	100	Normal
VCSW2K82	Offline	Not Waiting	0	Normal

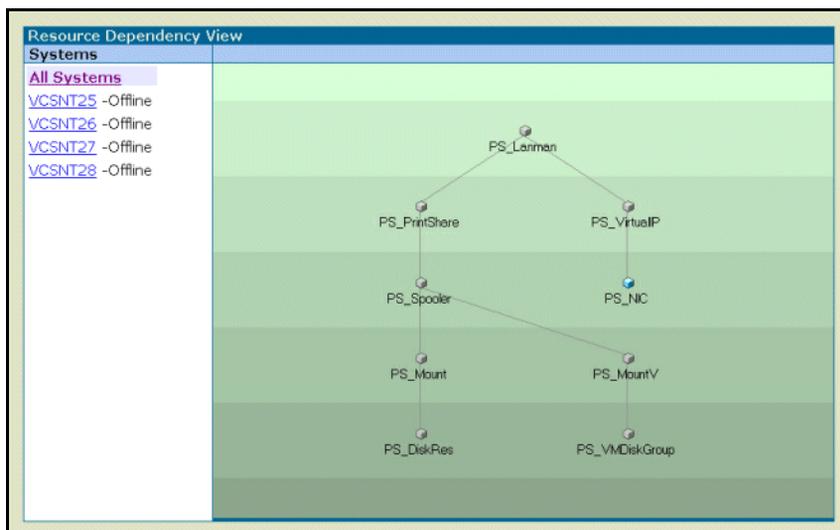
Recent Logs for this Object Current Time at Server : Dec 28, 2001 4:41:13 PM  
 No Logs for this object



## Resource Dependency Page

The Resource Dependency page displays a graph of resource dependencies within a service group. Dependencies between resources specify the order in which resources are brought online and taken offline. To view resource dependencies on a particular system, click the system name in the Systems list. To access a resource page, click the resource icon in the dependency graph.

- ✓ To access this page, click **Resource Graph** from the **Service Group** page. (Click Cluster Summary>All Groups>*service group*.)



## Installation and Configuration

The Web Console is installed with VCS. It works from a Web server on a system in the cluster. When the system is taken offline, the Web server is failed over to another system, which makes this GUI highly available.

A service group, ClusterService, monitors the Web server. The service group has an IP resource that creates a virtual IP address, and a VRTSWebApp resource that monitors the Web Console application running inside the Web server process. The Web Console is accessed using the virtual IP address, which enables continuous access when the Web Console fails over to another system.

If you choose to enable VCS Cluster Manager (Web Console) while installing VCS, the install program creates the ClusterService service group. Otherwise, you must create and configure the service group manually, as instructed below.

### Configuring the Web Console Manually

You must create and configure the ClusterService service group manually if you did not enable VCS Cluster Manager (Web Console) option while installing VCS. You can use Cluster Manager (Java Console) or the command-line interface to create and configure the service group.

1. Create a service group called ClusterService.
2. Add a resource of type IP to the service group. Name the resource **webip**. Configure the following attributes for the resource.

**Note:** You must configure *each* of the following attributes:

- **Address:** A virtual IP address to be assigned to VCS Cluster Manager (Web Console.) The GUI will be accessed using this IP address.
- **Device:** The name the public network card on the system from which the Web GUI will run. Device is defined as a local attribute for each system in the cluster.
- **NetMask:** The subnet to which the virtual IP address belongs.
- **Critical:** Set this attribute to **True** to make webip a critical resource.



3. Add a resource of type VRTSWebApp to the service group. Name the resource **VCSweb**. Configure the following attributes for the resource.

**Note:** You must configure *each* of the following attributes:

- **AppName:** Set to **vcs**.
  - **InstallDir:** Set to **/opt/VRTSweb/VERITAS**.
  - **TimeForOnline:** Set to **5**.
  - **Critical:** Set to **False**.
4. Link the VCSweb and webip resources, making VCSweb the parent resource.
  5. Enable both resources.
  6. Bring the ClusterService service group online. You can now access the Web Console at the URL `http://IP_alias:8181/vcs`. The variable `IP_alias` is the virtual IP address configured in the service group.

## Sample Configuration

```
group ClusterService (  
  SystemList = { vcsaix5, vcsaix6 }  
  AutoStartList = { vcsaix5, vcsaix6 }  
  OnlineRetryLimit = 3  
)  
  
IP webip (  
  Address = "162.39.9.85"  
  NetMask = "255.255.255.0"  
  Device = "en0"  
)  
  
VRTSWebApp VCSweb (  
  AppName = "vcs"  
  InstallDir = "/opt/VRTSweb/VERITAS"  
  TimeForOnline = 5  
  Critical = 0  
)  
  
VCSweb requires webip
```

---

## Before Using the Web Console

- ✓ Verify that the Web Console is installed on all systems in the cluster.
- ✓ Verify the ClusterService service group configuration. Verify that the service group is online. For more information, review “[Configuring the Web Console Manually](#)” on page 215.
- ✓ The Web Console requires an exclusive port. Verify that no other application is bound to the default VERITAS port, 8181.
- ✓ Install the Internet Explorer or Netscape Navigator browser, version 4.0 or later, on the system from which you will monitor and administer the cluster.
- ✓ Verify that Java is enabled for the browser.
- ✓ Verify that cookies are enabled for the browser.

---

**Note** Certain cluster operations are enabled or restricted depending on the privileges with which you log on to VCS. For information the specific privileges associated with VCS users, see [Chapter 5](#).

---

## Connecting to the Web Console

The Web Console can be accessed from the URL `http://IP_alias:8181/vcs`. The variable *IP\_alias* is the virtual IP address configured in the ClusterService service group, and the number 8181 is the default VERITAS Web port. *You must use a valid VCS user name and password to log on to the cluster.*



## Navigating the Web Console

The Web Console graphical user interface provides easy access to a cluster and its components through various methods of navigation, including trails, buttons, and links.

### Using Navigation Trails

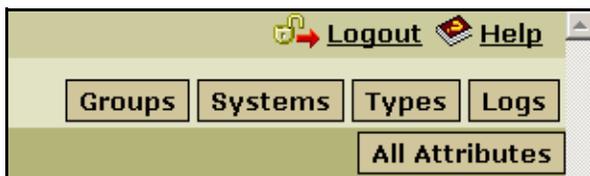
The Web Console follows a top-down navigational approach. The top left corner of each page displays a “breadcrumb trail” indicating the page’s position in the navigational hierarchy. The components of the trail are links to pages higher up in the hierarchy. For example, if you are on a Resource page, the navigation trail shows *cluster* → *All Groups* → *service group* → *resource*.



- ◆ Click the service group name to view information about the particular service group.
- ◆ Click **All Groups** to view information about all service groups.
- ◆ Click the cluster name to view general information about the cluster.

### Using Navigation Buttons

The top right corner of each page has buttons that link to other pages in the GUI. For example, click **Systems** to view the **All Systems** page.



### Using Information Links

The Web Console links some of its information to additional pages in the Web Console. For example, the All Systems page displays information about online and faulted groups on each system. These group names link to their respective pages.

## Refreshing the Web Console

The **Update** icon indicates the need to refresh the page when the information displayed is no longer current. Click the **Refresh Mode** icon to refresh the page and disable update notification. These icons are located in the top right corner of the Web console.



## Selecting the Refresh Mode

Select the appropriate refresh mode to refresh the data automatically or manually. The refresh mode icon alters its appearance according to the mode selected. The Web Console supports the following refresh modes.

-  **Auto.** Automatically updates the page when the data displayed is no longer current.
-  **Notify Only.** Informs you of the need to refresh the page. When there is an update, the Refresh icon  appears in place of the Update icon. Click the icon to update the page.
-  **Disable.** Disables update notification. Click the browser's Refresh button to refresh the page and retrieve current information.

You can change the refresh mode by clicking the Refresh Mode icon until you see the appropriate option.

## Understanding the Update Icon

The color of the Update icon indicates the state of the information on the page.

- ◆ A green icon indicates information on the page is current.
- ◆ A blinking orange icon indicates information on the page is not current and must be refreshed.
- ◆ A blue icon indicates the Web Console is connecting to the server.
- ◆ A red icon indicates the Web Console is disconnected from the server.
- ◆ A gray icon indicates update notification is disabled.

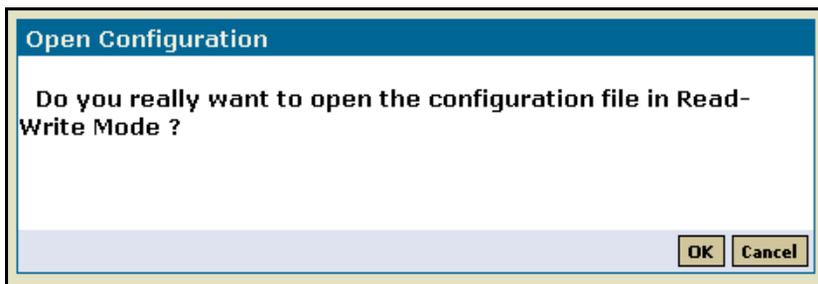


## Administering Clusters

The Web Console enables you to modify the VCS configuration. After opening the configuration, you can “dump” it (save it to disk).

### Opening the Configuration

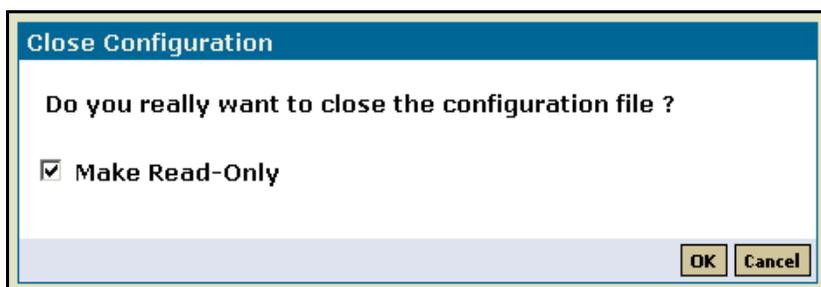
1. In the **Cluster Summary** page, click **Open Configuration**.



2. In the **Open Configuration** dialog box, click **OK**. The configuration becomes read/write which enables you to modify the configuration.

### Saving the Configuration

1. In the **Cluster Summary** page, click **Dump Configuration**.

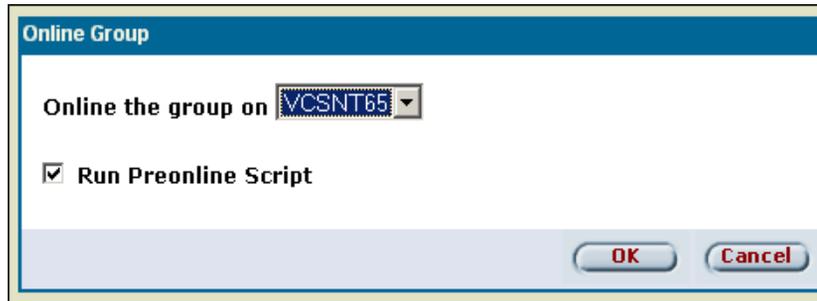


2. In the **Close Configuration** dialog box, click the **Make Read-Only** check box if you want to prevent any write operations to the configuration file.
3. Click **OK**.

# Administering Service Groups

## Bringing a Service Group Online

1. In the **Service Group** page (Cluster Summary>All Groups>*service group*), click **Online**. This opens the **Online Group** dialog box.

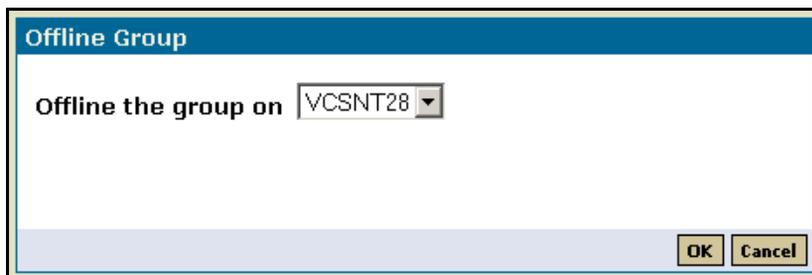


2. Click the system on which to bring the service group online.
3. To run PreOnline script, click the **Run PreOnline Script** check box.
4. Click **OK**.



## Taking a Service Group Offline

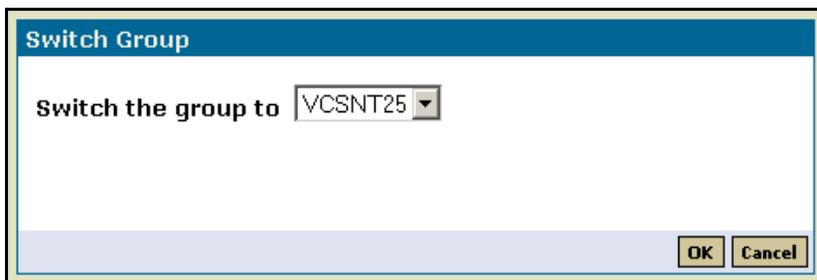
1. In the Service Group page (Cluster Summary>All Groups>*service group*), click **Offline**. This opens the **Offline Group** dialog box.



2. For parallel groups, click the system on which to take the service group offline.
3. Click **OK**.

## Switching a Service Group

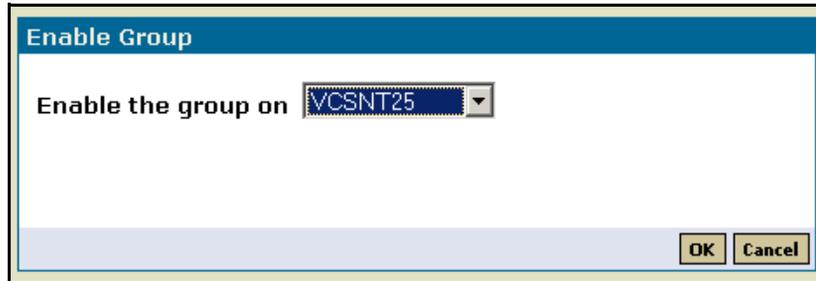
1. In the **Service Group** page (Cluster Summary>All Groups>*service group*), click **Switch**. This opens the **Switch Group** dialog box.



2. Click the system to switch the service group.
3. Click **OK**.

## Enabling a Service Group

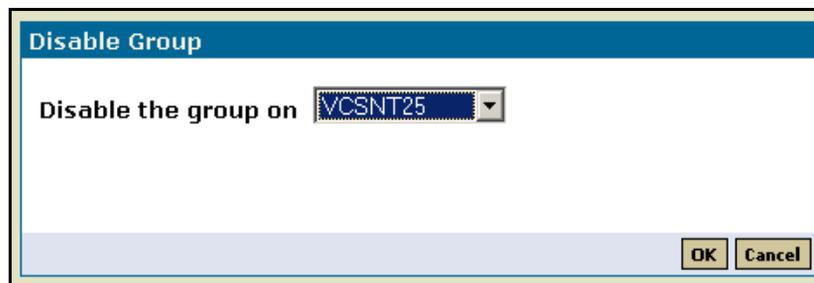
1. In the **Service Group** page (Cluster Summary>All Groups>*service group*), click **Enable**. This opens the **Enable Group** dialog box.



2. Click the system on which to enable the service group. To enable the service group on all systems, click **All Systems**.
3. Click **OK**.

## Disabling a Service Group

1. In the **Service Group** page (Cluster Summary>All Groups>*service group*), click **Disable**. This opens the **Disable Group** dialog box.

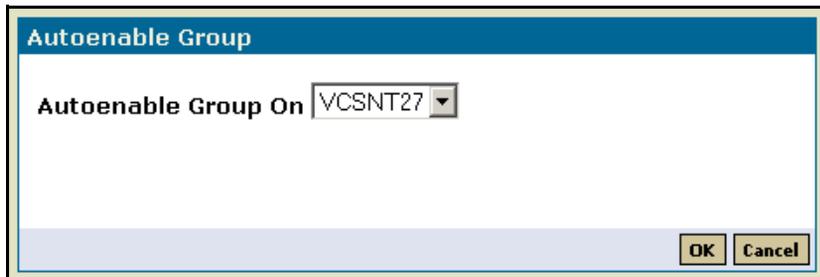


2. Click the system on which to disable the service group. To disable the service group on all systems, click **All Systems**.
3. Click **OK**.



## Autoenabling a Service Group

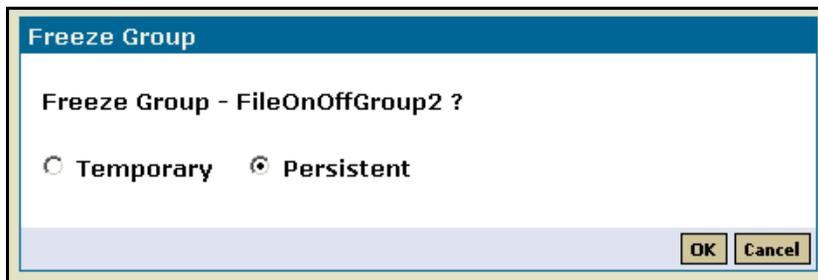
1. In the **Service Group** page (Cluster Summary>All Groups>*service group*), click **Autoenable**. This opens the **Autoenable Group** dialog box.



2. Click the system on which to autoenable the service group.
3. Click **OK**.

## Freezing a Service Group

1. In the **Service Group** page (Cluster Summary>All Groups>*service group*), click **Freeze**. This opens the **Freeze Group** dialog box.



2. Click the **Temporary** or **Persistent** option. The Persistent option enables the service group to retain its frozen state when the cluster is rebooted.
3. Click **OK**.

## Unfreezing a Service Group

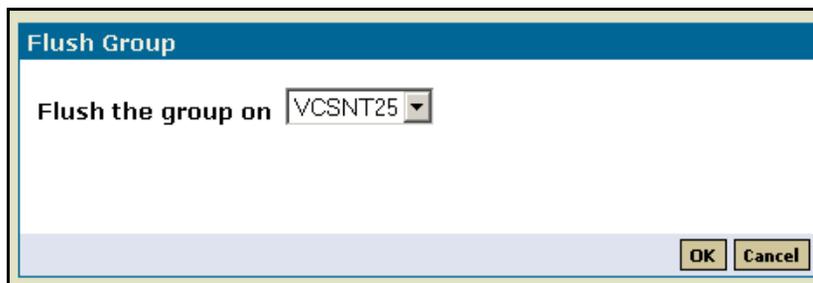
1. In the **Service Group** page (Cluster Summary>All Groups>*service group*), click **Unfreeze**. This opens the **Unfreeze Group** dialog box.



2. Click the **Temporary** or **Persistent** option, if applicable.
3. Click **OK**.

## Flushing a Service Group

1. In the **Service Group** page (Cluster Summary>All Groups>*service group*), click **Flush**. This opens the **Flush Group** dialog box.



2. Click the system on which to flush the service group.
3. Click **OK**.

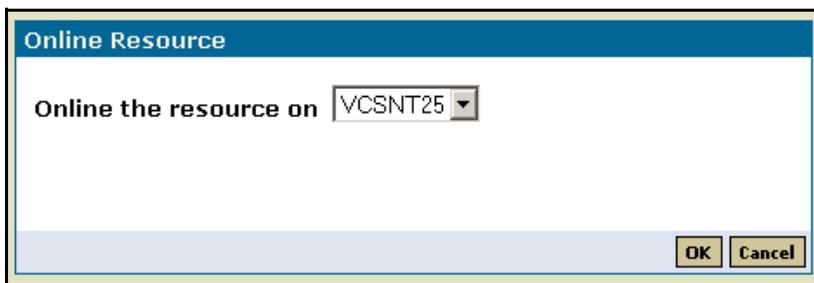


## Administering Resources

The Web Console enables you to perform a variety of operations through the Resource page. In the Service Group page (Cluster Summary>All Groups>*service group*), click the resource name in the Resources list.

### Bringing a Resource Online

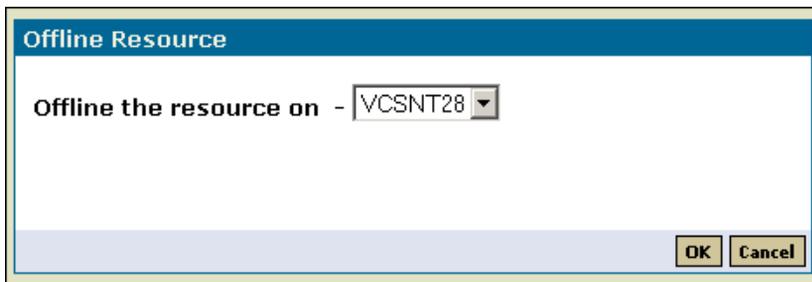
1. In the **Resource** page (Cluster Summary>All Groups>*service group*>*resource*), click **Online**. This opens the **Online Resource** dialog box.



2. Click the system on which to bring the resource online.
3. Click **OK**.

### Taking a Resource Offline

1. In the **Resource** page (Cluster Summary>All Groups>*service group*>*resource*), click **Offline**. This opens the **Offline Resource** dialog box.



2. Click the system on which to take the resource offline and click **OK**.

## Clearing a Resource

Clear a resource to remove a fault and make the resource available to go online. A fault is generated when a resource is taken offline unexpectedly.

1. In the **Resource** page (Cluster Summary>All Groups>*service group*>*resource*), click **Clear**. This opens the **Clear Resource** dialog box.

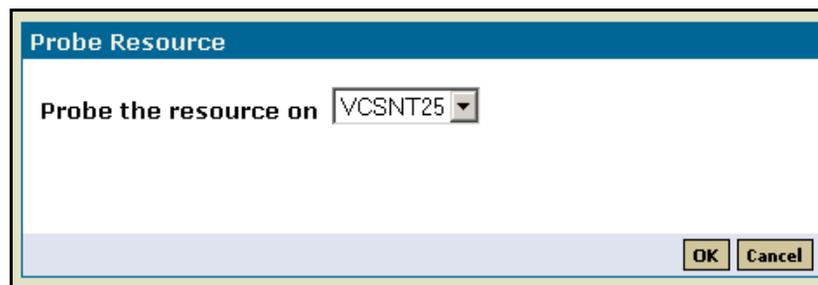


2. Click the system on which to clear the resource. To clear the resource on all systems, click **All Systems**.
3. Click **OK**.

## Probing a Resource

Probe a resource to verify that it is configured and ready to bring online.

1. In the **Resource** page (Cluster Summary>All Groups>*service group*>*resource*), click **Probe**. This opens the **Probe Resource** dialog box.



2. Click the system on which to probe the resource and click **OK**.

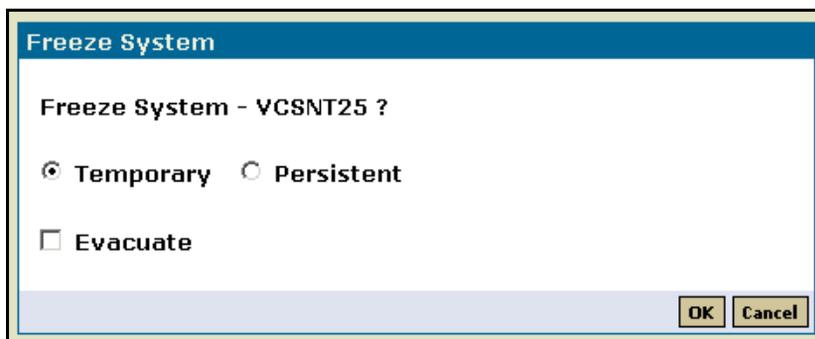


## Administering Systems

The Web Console enables you to freeze and unfreeze systems. In the System page (Cluster Summary>All System>*system*), freeze a system to stop all online and switch operations on the system.

### Freezing a System

1. In the **System** page (Cluster Summary>All Systems>*system*), click **Freeze**. This opens the **Freeze System** dialog box.

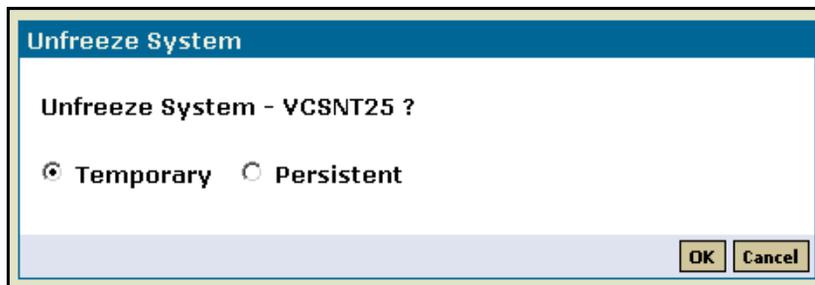


2. Click the **Persistent** or **Temporary** option. The Persistent option enables the freeze to be “remembered” when the cluster is rebooted.
3. Click the **Evacuate** check box, if required. This fails over the system’s active service groups to another system in the cluster before the freeze is enabled.
4. Click **OK**.

## Unfreezing a System

Thaw a frozen system to reenable online or switching operations.

1. In the **System** page (Cluster Summary>All Systems>*system*), click **Unfreeze**. This opens the **Unfreeze System** dialog box.



2. Click **OK**.



## Editing Attributes

The Web Console enables you to edit attributes of certain cluster objects, including service groups, systems, and resources. By default, the Web Console displays key attributes. To view the entire list of attributes associated with a cluster object, click **All Attributes**.

**Note** The configuration must be open (in read/write mode) to edit attributes. See “[Opening the Configuration](#)” on page 220 for instructions.

1. Go to the page of the object whose attributes you want to edit. For example, to edit system attributes, go to the **System** page. (Click Cluster Summary>All Systems>system.)
2. In the **Attributes** table, click the Edit icon (...) for the attribute.

Disk		
Attributes	Value	Edit
ArgList	DiskNo Signature AgentDebug	...
Operations	None	...
Restart Limit	0	...
Online Timeout	300	...
Offline Timeout	300	...
Monitor Timeout	60	...
Monitor Interval	60	...
Monitor (if Offline)	1	...
Log Level	error	...

3. In the **Edit Attribute** dialog box, enter the new value for the attribute and click **OK**.

- ◆ **To edit a scalar value**

Enter the value and click **OK**.

The screenshot shows a dialog box titled "FS\_NIC". It contains the following information:

- Attribute**: MaxTxErrInterval
- Dimension**: Scalar
- Description**: The number of consecutive monitor intervals for which the percentage of erroneous packets should not exceed MaxTxErrorPercentage. When this occurs, the NIC resource is declared offline. Default is 10 monitor intervals.
- Scope**: Global (with a "Make Local" button)
- Value**: 10 (in a text input field)
- Buttons: OK, Cancel

- ◆ **To edit an association**

Enter the key and the associated value. Click **Add** after entering each key-value pair and then click **OK**.

The screenshot shows a dialog box titled "FileOnOffGroup2". It contains the following information:

- Attribute**: SystemList
- Dimension**: Association
- Description**: List of systems on which the service group is configured to run and their priorities. Default = "" (none).
- Scope**: Global
- Key**: [input field]
- Value**: [input field]
- Buttons: Add, OK, Cancel

VCSNT25	4	<a href="#">Delete</a>
VCSNT26	3	<a href="#">Delete</a>
VCSNT27	2	<a href="#">Delete</a>
VCSNT28	1	<a href="#">Delete</a>



◆ **To edit a vector value**

Enter the value, click **Add** and then click **OK**.

The screenshot shows a dialog box titled "FS\_DiskRes" with a blue header. The main area has a light blue background and contains the following text:

**Attribute** : Signatures                      **Dimension** : Vector  
**Description** : An array specifying the signature of each SCSI disk.

**Scope:**                      Global                     

**Value**                      

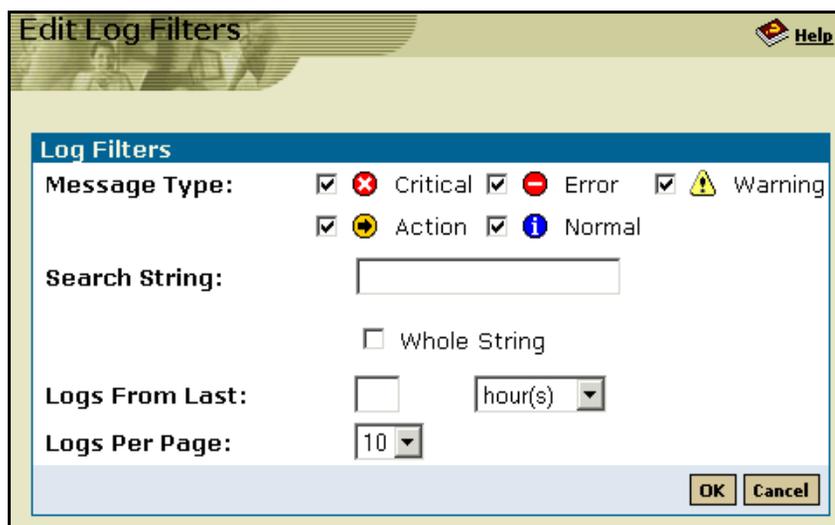
3q4447699	<a href="#">Delete</a>
23445788	<a href="#">Delete</a>
22332243	<a href="#">Delete</a>
455	<a href="#">Delete</a>
123	<a href="#">Delete</a>

## Customizing the Log Display

The Web Console enables you to customize the log display of messages generated by the VCS engine, HAD. In the Log Details page (click Logs from any page), you can set filter criteria to search and view messages.

### Editing the Log Display

1. In the **Log Details** page, click **Edit Filter**. This opens the **Edit Log Filters** dialog box.



2. Click the appropriate **Message Type** check boxes to customize the message display.
3. To search log messages, enter the search strings. Click the **Whole String** check box, if required.
4. To view log messages from a particular period, enter the value and click the time unit.
5. In the **Logs Per Page** drop-down list box, click the number of messages to display on each page.
6. Click **OK**.



## Resetting the Log Display

Use the Reset Filter to set the default settings for the log view. For example, if you customized the log view to only show critical and error messages using the Edit Filter feature, the Reset Filter feature sets the view to show all log messages.

The Reset Filter feature does not reset default settings for user preferences. This applies to the number of log messages viewed per page.

In the **Logs** page, click **Reset Filter** to reset the log display.





## Section IV Advanced Topics—What VCS Can Do

---

This section explains the latest enhancements to VCS functionality: workload balancing, notification, and event triggers. It also describes the impact VCS has on overall system performance, and includes methods on how to adjust VCS to meet specific configuration requirements. This section also includes a chapter on troubleshooting and recovery, which provides important information on topics such as handling network failure and deciphering common error messages.

Section IV includes the following chapters:

- ◆ **Chapter 9 “Service Group Workload Management” 237**
- ◆ **Chapter 10 “Notification” 259**
- ◆ **Chapter 11 “Event Triggers” 271**
- ◆ **Chapter 12 “VCS Performance Considerations” 281**
- ◆ **Chapter 13 “Troubleshooting and Recovery” 293**

## Managing Load from Service Group and System Levels

VCS includes a powerful load-balancing mechanism at the service group level. Known as “Service Group Workload Management,” this feature determines the optimal system on which to host an application during startup, or after an application or server faults. With Service Group Workload Management, you can statically associate system capacity and service group load within the main configuration file, `main.cf`, which is particularly useful when managing multiple service groups and systems.

VCS also maintains the functionality of dynamic system load, which can be used when service groups deviate heavily from their static load. Note that dynamic load is specified at the system level, meaning you can specify the load regardless of which service group is loading the system. This helps account for system load contributed by an application outside VCS of control.

## Deciding Startup and Failover Locations

Service Group Workload Management provides tools for making intelligent decisions about startup or failover locations, based on system capacity and finite resource availability. This feature is enabled when the service group attribute `FailOverPolicy` is set to “Load,” described on [page 238](#). The value of this attribute governs how VCS calculates the target system for failover. There are three possible values: `Priority`, `RoundRobin`, and `Load`.

### Priority

Priority policy is the most basic and is ideal for simple two-node clusters or small clusters with few service groups. With `FailOverPolicy` set to `Priority`, the system with the lowest priority in a running state is selected as the failover target. Priority is set in the `SystemList` attribute implicitly via ordering, such as `SystemList = { SystemA, SystemB }` or explicitly, such as `SystemList = { SystemA=0, SystemB=1 }`. Priority is the default behavior.



## RoundRobin

RoundRobin policy selects the system running the fewest service groups as the failover target. This is ideal for large clusters running many service groups with essentially the same server load characteristics (for example, similar databases or applications).

## Load

Load policy is the most flexible and powerful. It provides the framework for true server consolidation at the data center. Load policy comprises two components: system capacity and service group load, represented by the attributes Capacity and Load, respectively, and system limits and service group prerequisites, represented by the attributes Limits and Prerequisites, respectively. The following sections describe Load policy in detail.

### System Capacity and Service Group Load

The system attribute Capacity sets a fixed load-handling capacity for servers. You define this attribute with an arbitrary value. The service group attribute Load sets a fixed demand for service groups. You define this attribute based on application requirements. When a group is brought online, its Load value is subtracted from the Capacity value of the system to determine available capacity (represented by the attribute AvailableCapacity). The VCS engine, HAD, tracks the available capacity of all systems in the cluster. Note that a service group is considered online if it is online or partially online on a system.

When a failover must occur, HAD determines the system with the highest AvailableCapacity value and starts the service group on that system. During a failover involving multiple service groups, failover decisions are made serially to facilitate the proper load-based choice. When VCS selects a target node, available capacity for that node is reduced by the load of the service groups; that is, the value of the AvailableCapacity attribute is reduced. To fail over the remaining service groups, the reduced available capacity of the node is compared with other nodes. Bringing service groups online in parallel immediately follows.

System capacity is a *soft* restriction, meaning the value of the Capacity attribute can be less than zero. During a cascading failure scenario, the value of the AvailableCapacity attribute can be negative.

## Static Load versus Dynamic Load

Dynamic load is an integral component of the Service Group Workload Management framework. Typically, HAD sets remaining capacity with the function:

$$\text{AvailableCapacity} = \text{Capacity} - (\text{the sum of the online service groups' Load values}).$$

If the DynamicLoad attribute is specified with `hasys -load`, this value overrides calculated Load values with the function:

$$\text{AvailableCapacity} = \text{Capacity} - \text{DynamicLoad}$$

This enables finer control of system-loading values than estimated service group loading (static load). The downside is the user must set up and maintain a load-estimation package outside VCS. Note that DynamicLoad specified with `hasys -load` is subtracted from the attribute Capacity as an integer, not a percentage. For example, if a system Capacity value is 200 and the outside package determines the server is 80 percent loaded, the package informs VCS the DynamicLoad value is 160 (not 80). This is done by first querying VCS for the value of Capacity with `hasys -display`, then using the result of the query to calculate the actual load value to pass back in.

## Overload Warning

Overload warning provides the notification component of the Load policy. When a server sustains the preset load level (set by the attribute LoadWarningLevel, static or dynamically determined) for a preset time (set by the attribute LoadTimeThreshold), the LoadWarning trigger is initiated. For a full description of event management with triggers, see [Chapter 11](#). For details on the attributes cited above, see “[System Attributes](#)” on page 353.

The LoadWarning trigger is a user-defined script or application designed to carry out specific actions. It is invoked once, when system load exceeds the LoadWarningLevel for the LoadTimeThreshold. It is not invoked again until the LoadTimeCounter is reset.

Sample scripts provide simple operator warning on overload and methods to move or shut down groups, based on user-defined priority values. For example, if load on a server running a business-critical database reaches and stays above a user-defined threshold, operators are notified immediately. The LoadWarning trigger then scans the system for any service groups with a lower priority than the database (such as an internal HR application) and move the application to a less loaded system or shut down the application. The framework is completely flexible. The installer or user is free to implement any overload management scheme.



## Limits and Prerequisites

System limits and service group prerequisites increase capability of the Load policy and can be used in many ways. In the attribute Limits, you can set a list of finite resources available on a server, such as shared memory segments, semaphores, and others. For each service group you then define its prerequisites, which are used to manage application requirements. For example, a database may need three shared memory segments and ten semaphores. VCS Load policy first determines a subset of all systems meeting these criteria, and then chooses the least-loaded system from this set. In this way, an unloaded system that does not meet the prerequisites of a group is not chosen. As soon as the decision is made to bring a group online on a particular system, the values in the group's Prerequisites attribute are subtracted from the values in the system's Limits attribute.

When configuring the attributes Limits and Prerequisites, define the service group's prerequisites first, then the corresponding system limits. Note that each system can have a different limit, and that there is no cap on the number of group prerequisites and system limits. Group prerequisites and system limits can appear in any order. Not all groups are required to define all prerequisites, and not all systems are required to define all limits. If a system does not have defined limits for a particular system resource, then the default value of 0 is assumed. Similarly, when a group does not define prerequisites for particular system resource, the default value of 0 is assumed. However, all prerequisites specified in a group's Prerequisites attribute must be met on a system. If they are not, the service group cannot be brought online on the system.

System limits and group prerequisites can be used in many ways. In addition to hosting service groups on different systems, you can also configure the cluster N-to-1 or N-to-N configurations.

For example, if someone wants only one group online at a time on a system, add the following entries to the definition of each group and system:

```
Prerequisites = { GroupWeight = 1 }  
Limits = { GroupWeight = 1 }
```

In the example above, if there is a backup node, it can host one service group. If Limits = { GroupWeight = 3 } for the backup node, it can host three.

System limits and group prerequisites work independently of FailOverPolicy. Prerequisites are used to determine a subset of eligible systems on which a group can be started during failover or startup. When a list of systems is created, HAD then follows the configured FailOverPolicy.



## Capacity and Limits Together

Capacity and limits combined make a very powerful tool for determining the proper failover node. The system meeting the prerequisites and with the highest available capacity is always chosen. If multiple systems meet the prerequisites, and have the same values for the attribute `AvailableCapacity`, the system lexically first in the `SystemList` is chosen.

System limits is a hard value. This means a server is not chosen if it does not meet the prerequisites of the group. This cannot be overridden. Capacity is a soft limit. This means the system with the highest available capacity is chosen, even if this results in a negative `AvailableCapacity` value. Systems with available capacity of less than the percentage set by the attribute `LoadWarningLevel`, and those remaining at that load for longer than the time designated in the `LoadTimeThreshold` attribute, invoke the `LoadWarning` trigger described below.

## Additional Considerations

### System Zones

The attribute `SystemZones` provides a subset of systems to use in an initial failover decision. System zones are based on primary/backup server preferences. A service group tries to stay within its zone before choosing a host in another. For example, imagine a typical, three-tier application infrastructure with Web, application, and database servers. The application and database servers are configured in a single cluster. Using `SystemZones` requires a service group in the application zone to try to fail over to another application zone server if another is available. If not, the group fails over to the database zone, based on load and limits. In this configuration, excess capacity and limits on the database backend are essentially kept in reserve for the larger load of a database failover, while application servers handle the load of any groups in the application zone. During a cascading failure, excess capacity in the cluster remains available to any service group. The `SystemZones` feature allows fine-tuning of application failover decisions, yet retains the flexibility to fail anywhere in the cluster if necessary.



## Load-Based AutoStart

VCS 2.0 provides a method to determine where a service group comes online when the cluster initially starts. Administrators can set the `AutoStartPolicy` to `Load`, which allows the VCS engine, HAD, to determine the best system on which to start the groups. Service groups are placed in an `AutoStart` queue for load-based startup as soon as the group probes all running systems. As with failover, a subset of systems meet all prerequisites is created first. The system with the highest `AvailableCapacity` is then chosen from the subset.

Using `AutoStartPolicy = Load` and `SystemZones` enables the administrator to establish a list of preferred systems on which to initially run a group. As mentioned above, in a three-tier architecture, you would want application groups to start first in the application zone and database groups to start in the database zone.



## Sample Definitions and Configurations

### System and Service Group Definitions

The following main.cf example shows various Service Group Workload Management attributes in a system definition and a service group definition. For more information regarding the attributes cited below, see [Appendix B](#).

```
include "types.cf"
cluster SGWM-demo (
)

system LargeServer1 (
  Capacity = 200
  Limits = { ShrMemSeg=20, Semaphores=10, Processors=12 }
  LoadWarningLevel = 90
  LoadTimeThreshold = 600
)

group G1 (
  SystemList = { LargeServer1, LargeServer2, MedServer1,
                MedServer2 }
  SystemZones = { LargeServer1=0, LargeServer2=0,
                  MedServer1=1, MedServer2=1 }
  AutoStartPolicy = Load
  AutoStartList = { MedServer1, MedServer2 }
  FailOverPolicy = Load
  Load = 100
  Prerequisites = { ShrMemSeg=10, Semaphores=5, Processors=6 }
)
```



## Sample Configuration (Basic 4-Node Cluster)

```
include "types.cf"
cluster SGWM-demo

system Server1 (
  Capacity = 100
)

system Server2 (
  Capacity = 100
)

system Server3 (
  Capacity = 100
)

system Server4 (
  Capacity = 100
)

group G1 (
  SystemList = { Server1, Server2, Server3, Server4 }
  AutoStartPolicy = Load
  AutoStartList = { Server1, Server2, Server3, Server4 }
  FailOverPolicy = Load
  Load = 20

group G2 (
  SystemList = { Server1, Server2, Server3, Server4 }
  AutoStartPolicy = Load
  AutoStartList = { Server1, Server2, Server3, Server4 }
  FailOverPolicy = Load
  Load = 40

group G3 (
  SystemList = { Server1, Server2, Server3, Server4 }
  AutoStartPolicy = Load
  AutoStartList = { Server1, Server2, Server3, Server4 }
  FailOverPolicy = Load
  Load = 30
```

```
group G4 (  
  SystemList = { Server1, Server2, Server3, Server4 }  
  AutoStartPolicy = Load  
  AutoStartList = { Server1, Server2, Server3, Server4 }  
  FailOverPolicy = Load  
  Load = 10  
  
group G5 (  
  SystemList = { Server1, Server2, Server3, Server4 }  
  AutoStartPolicy = Load  
  AutoStartList = { Server1, Server2, Server3, Server4 }  
  FailOverPolicy = Load  
  Load = 50  
  
group G6 (  
  SystemList = { Server1, Server2, Server3, Server4 }  
  AutoStartPolicy = Load  
  AutoStartList = { Server1, Server2, Server3, Server4 }  
  FailOverPolicy = Load  
  Load = 30  
  
group G7 (  
  SystemList = { Server1, Server2, Server3, Server4 }  
  AutoStartPolicy = Load  
  AutoStartList = { Server1, Server2, Server3, Server4 }  
  FailOverPolicy = Load  
  Load = 20  
  
group G8 (  
  SystemList = { Server1, Server2, Server3, Server4 }  
  AutoStartPolicy = Load  
  AutoStartList = { Server1, Server2, Server3, Server4 }  
  FailOverPolicy = Load  
  Load = 40
```



## AutoStart Operation

As mentioned in “[Load-Based AutoStart](#),” groups are placed in a queue when they are fully probed on all systems. For the purposes of this example, assume the groups probe in the same order they are described, G1 through G8.

Group G1 chooses the system with the highest value in the AvailableCapacity attribute. All are equal, so G1 starts on Server1 because this server is canonically first. Groups G2–G4 follow on Server2–Server4. With the startup decisions made for the initial four groups, the cluster resembles:

```
Server1
    AvailableCapacity=80
Server2
    AvailableCapacity=60
Server3
    AvailableCapacity=70
Server4
    AvailableCapacity=90
```

As the next groups come online, group G5 starts on Server4 because this server has the highest AvailableCapacity value. Group G6 then starts on Server1 with AvailableCapacity of 80. Group G7 comes online on Server3 with AvailableCapacity of 70. G8 comes online on Server2 with AvailableCapacity of 60.

## Normal Operation

The final cluster configuration (assuming the original queue of groups G1–G8) resembles:

```
Server1
    AvailableCapacity=50
    (groups G1 and G6)
Server2
    AvailableCapacity=20
    (groups G2 and G8)
Server3
    AvailableCapacity=50
    (groups G3 and G7)
```



#### Server4

AvailableCapacity=40  
(groups G4 and G5)

In this configuration, Server2 fires the LoadWarning trigger after the default 600 seconds because it is at default LoadWarningLevel of 80 percent.

### Failure Scenario

In the first failure scenario, Server4 fails. This immediately queues groups G4 and G5 for failure decision. Group G4 chooses Server1 because this server and Server3 have AvailableCapacity of 50 and Server1 is canonically first. Group G5 then goes online on Server3. Remember that failure decisions are made *serially*, not actual online and offline operations. The serializing of the failover choice allows complete load-based control and adds less than one second to total failover time.

Following the first failure, the configuration now resembles:

#### Server1

AvailableCapacity=40  
(groups G1, G6, and G4)

#### Server2

AvailableCapacity=20  
(groups G2 and G8)

#### Server3

AvailableCapacity=0  
(groups G3, G7, and G5)

In this configuration, Server3 fires the LoadWarning trigger to notify the administrator that the server is overloaded. The administrator could then switch group G7 to Server1 to balance load across groups G1 and G3. When Server4 is repaired, it rejoins the cluster with AvailableCapacity of 100. Further failover is sent to Server4.



## Cascading Failure Scenario

Even if Server4 was not immediately repaired, further failures are possible. For this example, assume Server3 now fails. Group G3 immediately chooses Server1, group G5 chooses Server2, and group G7 chooses Server1. This results in the following configuration:

Server1

AvailableCapacity=-10

(groups G1, G6, G4, G3, and G7)

Server2

AvailableCapacity=-30

(groups G2, G8, and G5)

In this example, we see how Capacity is a soft limit and can go below zero. Server1 will fire the LoadWarning trigger to notify the administrator that the server is overloaded.

## Sample Configuration (Complex 4-Node Cluster)

The following example shows the main configuration file, `main.cf`, of a four-node cluster using multiple system capacities and various limits. The cluster consists of two large enterprise servers (`LargeServer1` and `LargeServer2`) and two medium servers (`MedServer1` and `MedServer2`). There are four service groups, `G1`–`G4`, with various loads and prerequisites. Groups `G1` and `G2` are database applications with specific shared memory and semaphore requirements. Groups `G3` and `G4` are middle-tier applications with no specific memory or semaphores requirements, they simply add load to a system.

```
include "types.cf"
cluster SGWM-demo (
)

system LargeServer1 (
  Capacity = 200
  Limits = { ShrMemSeg=20, Semaphores=10, Processors=12 }
  LoadWarningLevel = 90
  LoadTimeThreshold = 600
)

system LargeServer2 (
  Capacity = 200
  Limits = { ShrMemSeg=20, Semaphores=10, Processors=12 }
  LoadWarningLevel=70
  LoadTimeThreshold=300
)

system MedServer1 (
  Capacity = 100
  Limits = { ShrMemSeg=10, Semaphores=5, Processors=6 }
)

system MedServer2 (
  Capacity = 100
  Limits = { ShrMemSeg=10, Semaphores=5, Processors=6 }
)
```



```
group G1 (
SystemList = { LargeServer1, LargeServer2, MedServer1, MedServer2 }
SystemZones = { LargeServer1=0, LargeServer2=0, MedServer1=1,
    MedServer2=1 }
AutoStartPolicy = Load
AutoStartList = { LargeServer1, LargeServer2 }
FailOverPolicy = Load
Load = 100
Prerequisites = { ShrMemSeg=10, Semaphores=5, Processors=6 }
)

group G2 (
SystemList = { LargeServer1, LargeServer2, MedServer1, MedServer2 }
SystemZones = { LargeServer1=0, LargeServer2=0, MedServer1=1,
    MedServer2=1 }
AutoStartPolicy = Load
AutoStartList = { LargeServer1, LargeServer2 }
FailOverPolicy = Load
Load = 100
Prerequisites = { ShrMemSeg=10, Semaphores=5, Processors=6 }
)

group G3 (
SystemList = { LargeServer1, LargeServer2, MedServer1, MedServer2 }
SystemZones = { LargeServer1=0, LargeServer2=0, MedServer1=1,
    MedServer2=1 }
AutoStartPolicy = Load
AutoStartList = { MedServer1, MedServer2 }
FailOverPolicy = Load
Load = 30
)

group G4 (
SystemList = { LargeServer1, LargeServer2, MedServer1, MedServer2 }
SystemZones = { LargeServer1=0, LargeServer2=0, MedServer1=1,
    MedServer2=1 }
AutoStartPolicy = Load
AutoStartList = { MedServer1, MedServer2 }
FailOverPolicy = Load
Load = 20
)
```

## AutoStart Operation

Based on the preceding main.cf example, the AutoStart sequence resembles:

G1 - LargeServer1

G2 - LargeServer2

G3 - MedServer1

G4 - MedServer2

All groups begin a probe sequence when the cluster starts. Groups G1 and G2 have an AutoStartList of LargeServer1 and LargeServer2. When these groups probe, they are queued to go online on one of these servers, based on highest AvailableCapacity value. Assuming G1 probes first, it chooses LargeServer1 because LargeServer1 and LargeServer2 both have an AvailableCapacity of 200, and LargeServer1 is lexically first. The same sequence occurs with groups G3 and G4 determining which server to choose between MedServer1 and MedServer2.

## Normal Operation

Based on the example beginning on [page 249](#), the final configuration resembles:

LargeServer1

AvailableCapacity=100

CurrentLimits={ ShrMemSeg=10, Semaphores=6, Processors=6 }

LargeServer2

AvailableCapacity=100

CurrentLimits={ ShrMemSeg=10, Semaphores=5, Processors=6 }

MedServer1

AvailableCapacity=70

CurrentLimits={ ShrMemSeg=10, Semaphores=5, Processors=6 }

MedServer2

AvailableCapacity=80

CurrentLimits={ ShrMemSeg=10, Semaphores=5, Processors=6 }



## Failure Scenario

In this failure example, assume system LargeServer2 fails. The VCS engine, HAD, first scans all available systems in group G2's SystemList that are grouped in the same SystemZones as the server on which the group was running. It then creates a subset of systems meeting the group's prerequisites. In this case, LargeServer1 meets all required Limits. Group G2 is brought online on LargeServer1. This results in the following configuration:

LargeServer1

AvailableCapacity=0

CurrentLimits={ ShrMemSeg=0, Semaphores=0, Processors=0 }

MedServer1

AvailableCapacity=70

CurrentLimits={ ShrMemSeg=10, Semaphores=5, Processors=6 }

MedServer2

AvailableCapacity=80

CurrentLimits={ ShrMemSeg=10, Semaphores=5, Processors=6 }

After 10 minutes, (LoadTimeThreshold = 600) the LoadWarning trigger on LargeServer1 fires because the LoadWarningLevel exceeds 90 percent.

## Cascading Failure Scenario

In this scenario, further failure of any system can be tolerated because each system has sufficient Limits remaining to accommodate the service group running on the peer. If a failure occurs with MedServer1 or MedServer2, the opposite MedServer is chosen because groups have MedServer1 and MedServer2 designated in their SystemZone attribute.

If a failure occurred on LargeServer1 while running two groups (LargeServer2 remains offline), the failover of the two groups is serialized for the decision process. In this case, no systems exist in the database zone. The first group, canonically G1, chooses MedServer2 because this server meets the required Limits and has the highest value in its AvailableCapacity attribute. Group G2 automatically chooses MedServer1 because it is the only remaining system that meets the required Limits.

## Sample Configuration (Server Consolidation)

The following example shows the main.cf configuration file for a complex eight-node cluster running multiple applications and several large databases. The database servers, LargeServer1, LargeServer2, and LargeServer3, are enterprise systems. The middle-tier servers running multiple applications are MedServer1, MedServer2, MedServer3, MedServer4, and MedServer5.

```
include "types.cf"
cluster SGWM-demo (
)

system LargeServer1 (
  Capacity = 200
  Limits = { ShrMemSeg=15, Semaphores=30, Processors=18 }
  LoadWarningLevel = 80
  LoadTimeThreshold = 900
)

system LargeServer2 (
  Capacity = 200
  Limits = { ShrMemSeg=15, Semaphores=30, Processors=18 }
  LoadWarningLevel=80
  LoadTimeThreshold=900
)

system LargeServer3 (
  Capacity = 200
  Limits = { ShrMemSeg=15, Semaphores=30, Processors=18 }
  LoadWarningLevel=80
  LoadTimeThreshold=900
)

system MedServer1 (
  Capacity = 100
  Limits = { ShrMemSeg=5, Semaphores=10, Processors=6 }
)

system MedServer2 (
  Capacity = 100
  Limits = { ShrMemSeg=5, Semaphores=10, Processors=6 }
)

system MedServer3 (
  Capacity = 100
  Limits = { ShrMemSeg=5, Semaphores=10, Processors=6 }
)
```



```

system MedServer4 (
  Capacity = 100
  Limits = { ShrMemSeg=5, Semaphores=10, Processors=6 }
)
system MedServer5 (
  Capacity = 100
  Limits = { ShrMemSeg=5, Semaphores=10, Processors=6 }
)

group Database1 (
  SystemList = { LargeServer1, LargeServer2, LargeServer3,
    MedServer1, MedServer2, MedServer3, MedServer4, MedServer5 }
  SystemZones = { LargeServer1=0, LargeServer2=0, LargeServer3=0,
    MedServer1=1, MedServer2=1, MedServer3=1, MedServer4=1,
    MedServer5=1 }
  AutoStartPolicy = Load
  AutoStartList = { LargeServer1, LargeServer2, LargeServer3 }
  FailOverPolicy = Load
  Load = 100
  Prerequisites = { ShrMemSeg=5, Semaphores=10, Processors=6 }
)

group Database2 (
  SystemList = { LargeServer1, LargeServer2, LargeServer3,
    MedServer1, MedServer2, MedServer3, MedServer4, MedServer5 }
  SystemZones = { LargeServer1=0, LargeServer2=0, LargeServer3=0,
    MedServer1=1, MedServer2=1, MedServer3=1, MedServer4=1,
    MedServer5=1 }
  AutoStartPolicy = Load
  AutoStartList = { LargeServer1, LargeServer2, LargeServer3 }
  FailOverPolicy = Load
  Load = 100
  Prerequisites = { ShrMemSeg=5, Semaphores=10, Processors=6 }
)

group Database3 (
  SystemList = { LargeServer1, LargeServer2, LargeServer3,
    MedServer1, MedServer2, MedServer3, MedServer4, MedServer5 }
  SystemZones = { LargeServer1=0, LargeServer2=0, LargeServer3=0,
    MedServer1=1, MedServer2=1, MedServer3=1, MedServer4=1,
    MedServer5=1 }
  AutoStartPolicy = Load
  AutoStartList = { LargeServer1, LargeServer2, LargeServer3 }
  FailOverPolicy = Load
  Load = 100
  Prerequisites = { ShrMemSeg=5, Semaphores=10, Processors=6 }
)

```



```
group Application1 (
  SystemList = { LargeServer1, LargeServer2, LargeServer3,
    MedServer1, MedServer2, MedServer3, MedServer4, MedServer5 }
  SystemZones = { LargeServer1=0, LargeServer2=0, LargeServer3=0,
    MedServer1=1, MedServer2=1, MedServer3=1, MedServer4=1,
    MedServer5=1 }
  AutoStartPolicy = Load
  AutoStartList = { MedServer1, MedServer2, MedServer3, MedServer4,
    MedServer5 }
  FailOverPolicy = Load
  Load = 50
)

group Application2 (
  SystemList = { LargeServer1, LargeServer2, LargeServer3,
    MedServer1, MedServer2, MedServer3, MedServer4, MedServer5 }
  SystemZones = { LargeServer1=0, LargeServer2=0, LargeServer3=0,
    MedServer1=1, MedServer2=1, MedServer3=1, MedServer4=1,
    MedServer5=1 }
  AutoStartPolicy = Load
  AutoStartList = { MedServer1, MedServer2, MedServer3, MedServer4,
    MedServer5 }
  FailOverPolicy = Load
  Load = 50
)

group Application3 (
  SystemList = { LargeServer1, LargeServer2, LargeServer3,
    MedServer1, MedServer2, MedServer3, MedServer4, MedServer5 }
  SystemZones = { LargeServer1=0, LargeServer2=0, LargeServer3=0,
    MedServer1=1, MedServer2=1, MedServer3=1, MedServer4=1,
    MedServer5=1 }
  AutoStartPolicy = Load
  AutoStartList = { MedServer1, MedServer2, MedServer3, MedServer4,
    MedServer5 }
  FailOverPolicy = Load
  Load = 50
)
```



```
group Application4 (  
  SystemList = { LargeServer1, LargeServer2, LargeServer3,  
    MedServer1, MedServer2, MedServer3, MedServer4, MedServer5 }  
  SystemZones = { LargeServer1=0, LargeServer2=0, LargeServer3=0,  
    MedServer1=1, MedServer2=1, MedServer3=1, MedServer4=1,  
    MedServer5=1 }  
  AutoStartPolicy = Load  
  AutoStartList = { MedServer1, MedServer2, MedServer3, MedServer4,  
    MedServer5 }  
  FailOverPolicy = Load  
  Load = 50  
)  
  
group Application5 (  
  SystemList = { LargeServer1, LargeServer2, LargeServer3,  
    MedServer1, MedServer2, MedServer3, MedServer4, MedServer5 }  
  SystemZones = { LargeServer1=0, LargeServer2=0, LargeServer3=0,  
    MedServer1=1, MedServer2=1, MedServer3=1, MedServer4=1,  
    MedServer5=1 }  
  AutoStartPolicy = Load  
  AutoStartList = { MedServer1, MedServer2, MedServer3, MedServer4,  
    MedServer5 }  
  FailOverPolicy = Load  
  Load = 50  
)
```

### AutoStart Operation

Based on the preceding main.cf example, the AutoStart sequence resembles:

- Database1 - LargeServer1
- Database2 - LargeServer2
- Database3 - LargeServer3
- Application1 - MedServer1
- Application2 - MedServer2
- Application3 - MedServer3
- Application4 - MedServer4
- Application5 - MedServer5

## Normal Operation

Based on the example beginning on [page 253](#), the final configuration resembles:

LargeServer1

AvailableCapacity=100

CurrentLimits={ ShrMemSeg=10, Semaphores=20, Processors=12 }

LargeServer2

AvailableCapacity=100

CurrentLimits={ ShrMemSeg=10, Semaphores=20, Processors=12 }

LargeServer3

AvailableCapacity=100

CurrentLimits={ ShrMemSeg=10, Semaphores=20, Processors=12 }

MedServer1

AvailableCapacity=50

CurrentLimits={ ShrMemSeg=5, Semaphores=10, Processors=6 }

MedServer2

AvailableCapacity=50

CurrentLimits={ ShrMemSeg=5, Semaphores=10, Processors=6 }

MedServer3

AvailableCapacity=50

CurrentLimits={ ShrMemSeg=5, Semaphores=10, Processors=6 }

MedServer4

AvailableCapacity=50

CurrentLimits={ ShrMemSeg=5, Semaphores=10, Processors=6 }

MedServer5

AvailableCapacity=50

CurrentLimits={ ShrMemSeg=5, Semaphores=10, Processors=6 }



## Failure Scenario

The preceding configuration is a good example of FailOverPolicy=Load and SystemZones. The database zone (system zone 0) can handle a maximum of two failures. Each server has adequate Limits to support a maximum of three database service groups (with an expected performance drop with all groups running on one server). Similarly, the application zone has excess capacity built into each server.

Note the servers running the application groups specify Limits to support one database, even though the application groups do not run prerequisites. This allows a database to fail across system zones if necessary, and run on the server with the least-loaded application zone.

In the following example, LargeServer3 fails. The VCS engine, HAD, first scans all available systems in Database2's SystemList that are grouped in the same SystemZones as the server on which the group was running. It then creates a subset of systems meeting the group's prerequisites. In this case, LargeServer1 and LargeServer2 meet the required Limits. Database1 is brought online on LargeServer1. This results in the following configuration:

LargeServer1

AvailableCapacity=0

CurrentLimits={ ShrMemSeg=5, Semaphores=10, Processors=6 }

LargeServer2

AvailableCapacity=100

CurrentLimits={ ShrMemSeg=10, Semaphores=15, Processors=12 }

In this scenario, further failure of any system can be tolerated because each system has sufficient Limits remaining to accommodate the service group running on the peer.

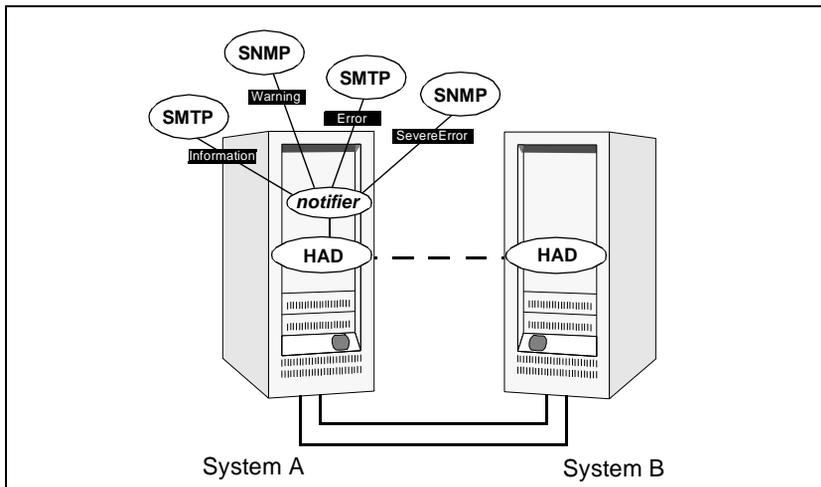
## Cascading Failure Scenario

If the performance of a specific database is unacceptable with two database groups running on a single server (or three following a second failure), the SystemZones policy can help expedite performance. Failing a database group into the application zone has the effect of resetting its preferred zone. For example, in the above scenario Database1 was moved to LargeServer1. The administrator could reconfigure the application zone to move two application groups to a single system. The database application can then be switched to the empty application server (MedServer1–MedServer5), which would put Database1 in Zone1 (application zone). If a failure occurs in Database1, the group selects the least-loaded server in the application zone meeting its prerequisites.

VCS provides a method for notifying the administrator of important events such as a resource or system fault. VCS includes a “notifier” component, which consists of the notifier process and the hanotify utility.

## How It Works

As illustrated below, the notifier process receives notification from HAD, formats the notification, then, according to the configuration, generates an SNMP (V2) trap or sends an email to the designated recipient, or both. There are four severity levels: SevereError, Error, Warning, and Information. SevereError indicates the highest severity level, Information the lowest.



SNMP traps sent by VCS are forwarded to the SNMP console. Typically, traps are predefined for events such as service group or resource faults. The hanotify utility enables you to send additional traps, apart from those sent by HAD.



## Event Messages and Severity Levels

When the VCS engine, HAD, starts up, it is initially configured to queue all messages as Information, the lowest severity level. However, when notifier connects to VCS, the severity communicated by notifier to HAD is one of the following, depending on which is the lowest:

- ◆ lowest severity for SNMP options
- ◆ lowest severity for SMTP options

Because notifier communicates the severity to HAD, HAD does not queue unnecessary messages. Also because HAD queues all messages as the lowest severity level until notifier connects to it (regardless of the actual severity) no messages are lost.

If notifier is started without specifying a severity level for the SNMP console or SMTP recipients, notifier assumes the default severity level Warning and communicates that to HAD.

For example, if the following severities are specified for notifier:

- ◆ Warning for email recipient 1
- ◆ Error for email recipient 2
- ◆ SevereError for SNMP console

Notifier communicates the minimum severity, Warning, to HAD, which then queues all messages labeled severity level Warning and greater.

Notifier ensures the recipient gets only the messages that he or she has been designated to receive (according to the specified severity level). However, until notifier communicates the specifications to HAD, HAD stores all messages, because it does not know the severity the user has specified. This prevents messages from being lost between the time HAD stores them and notifier communicates the specifications to HAD.

## Persistent and Replicated Message Queue

VCS includes a sophisticated mechanism for maintaining event messages that ensures messages are not lost. On each node, VCS queues messages to be sent to the notifier process. This queue is guaranteed persistent as long as VCS is running and the contents of this queue remain the same on each node. Therefore, if the group with notifier configured as a resource fails on one of the nodes, notifier is failed over to another node in the cluster. Because the message queue is guaranteed to be consistent and replicated across nodes, notifier can resume message delivery from where it left off after it fails over to the new node.

## How HAD Deletes Messages

The VCS engine, HAD, stores messages to be sent to notifier. These messages are deleted by HAD under the following conditions:

- ◆ The message has been in the queue for one hour and notifier is unable to deliver the message to the recipient. (This also means, that until notifier connects to HAD, messages are stored permanently in the queue until one of the following conditions are met.)  
*or*
- ◆ The message queue is full and to make room for the latest message, the earliest message is deleted.  
*or*
- ◆ VCS receives a message acknowledgement from notifier when notifier has delivered the message to at least one designated recipient. For example, if two SNMP consoles and two email recipients are designated, and notifier can send the message to only one email recipient because the other three were configured incorrectly, notifier sends an acknowledgement to VCS, regardless that the message reached only one of the four recipients. Error messages are also printed to the log files when delivery errors occur.



## Notification Components

### The notifier Process

The notifier process configures how messages are received from VCS and how they are delivered to SNMP consoles and SMTP servers. Using notifier, you can specify notification based on the severity level of the events generating the messages. You can also specify the size of the VCS message queue, which is 30 by default.

When started from the command line, notifier is a process that VCS does not control. For best results, use the NotifierMngr agent bundled with VCS to configure notifier as part of a highly available service group, which can then be monitored, brought online, and taken offline. For information on how to configure NotifierMngr, see the *VERITAS Cluster Server Bundled Agents Reference Guide*. Note that notifier must be configured in a failover group, not parallel, because only one instance of notifier runs in the entire cluster.

Notifier enables you to specify configurations for the SNMP manager and SMTP server, including machine names, ports, community IDs, and recipients' email addresses. You can specify more than one manager or server, and the severity level of messages sent to each.

### Example of notifier Command

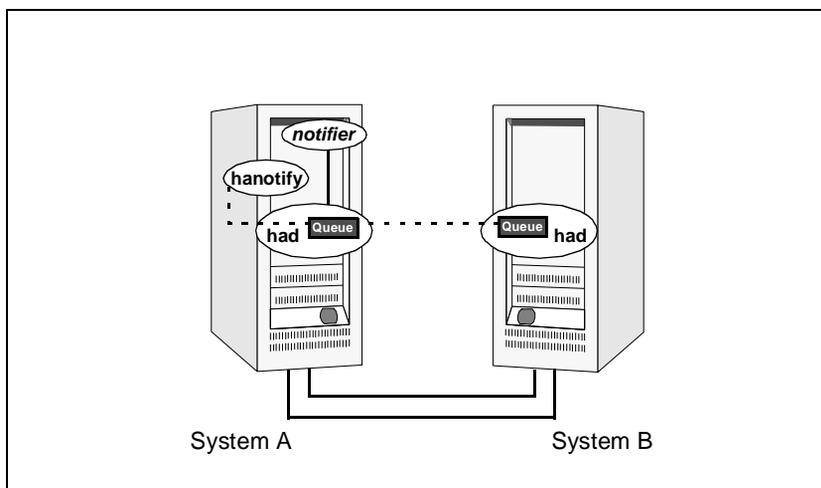
```
# notifier -s m=north -s m=south,p=2000,l=Error,c=your_company  
-t m=north,e="abc@your_company.com",l=SevereError
```

In this example, notifier:

- Sends all level SNMP traps to *north* at the default SNMP port and community value *public*.
- Sends Warning traps to *north*.
- Sends Error and SevereError traps to *south* at *port 2000* and community value *your\_company*.
- Sends SevereError email messages to *north* as SMTP server at default port and to email recipient *abc@your\_company.com*.

## The hanotify Utility

The hanotify utility enables you to construct user-defined messages. These messages are then forwarded by hanotify to HAD, which in turn stores them in its internal message queue. Along with other messages, user-defined messages are also forwarded to the notifier process for delivery to email recipients, SNMP consoles, or both.



### Example of hanotify Command

```
# hanotify -i "1.3.6.1.4.1.1302.3.8.10.27.2" -l Error -n
gcm -T 6 -t 1.1 -o 1 -s sys1 -L London -p sys2 -P
Paris -c site1 -C 6 -O admin -m "site1 is down"
```

In this example, the number 1.3.6.1.4.1.1302.3.8.10.27.2 is the OID for the message being sent. Because it is a user-defined message, VCS has no way of knowing the OID associated with the SNMP trap corresponding to this message so the user must provide it.

The other parameters to hanotify specify the message is severity level Error, the site is running GCM version 1.1. The systems affected are sys1 and sys2, which are located in London and Paris and compose site 1.



## List of Events and Traps

The tables below specify which events generate traps, email notification, or both. Note that “SevereError” indicates the highest severity level, “Information” the lowest.

### Clusters

---

Message	Severity Level	Description
User has logged on to VCS.	Information	A user log on has been recognized because a user logged on via Cluster Manager, or because a ha.xxx command was invoked.

---

### Agents

---

Message	Severity Level	Description
Agent is restarting	Information	VCS is restarting the agent.
Agent is faulted.	Warning	The agent has faulted on one node in the cluster.

---



## Resources

Message	Severity Level	Description
Resource state is unknown.	Warning	VCS cannot identify the state of the resource.
Resource monitoring has timed out.	Warning	Monitoring mechanism for the resource has timed out.
Resource is not going offline.	Warning	VCS cannot take the resource offline.
Resource is being restarted by agent.	Information	The resource is being restarted by its agent.
Resource went online by itself.	Warning (not for first probe)	The resource was brought online on its own.
Resource has faulted.	Error	Self-explanatory.



## Systems

---

Message	Severity Level	Description
VCS is up on the first node in the cluster.	Error	Self-explanatory.
VCS is being restarted by <code>hashadow</code> .	Warning	Self-explanatory.
VCS is in jeopardy.	Warning	One node running VCS is in jeopardy.
VCS has faulted.	Information	Self-explanatory.
A node running VCS has joined cluster.	Information	Self-explanatory.
VCS has exited manually.	Information	VCS has exited gracefully from one node on which it was previously running.
VCS is up but is not in the cluster.	Information	VCS is running on one node but the node is not visible.

---

## Service Groups

---

Message	Severity Level	Description
Service group is online	Information	Self-explanatory.
Service group is offline.	Information	Self-explanatory.
Service group is autodisabled.	Information	VCS has autodisabled the specified group because one node exited the cluster.
Service group has faulted.	SevereError	Self-explanatory.
Service group has faulted and cannot be failed over anywhere.	Error	Specified service group has faulted on all nodes where group could be brought online, and there are no nodes to which the group can fail over.
Service group is restarting.	Information	Self-explanatory.
Service group is being switched.	Information	The service group is being taken offline on one node and being brought online on another.

<b>Message</b>	<b>Severity Level</b>	<b>Description</b>
Service group concurrency violation.	SevereError	A failover service group has become online on more than one node in the cluster.
Service group restarting in response to persistent resource going online.	Information	Self-explanatory.

---



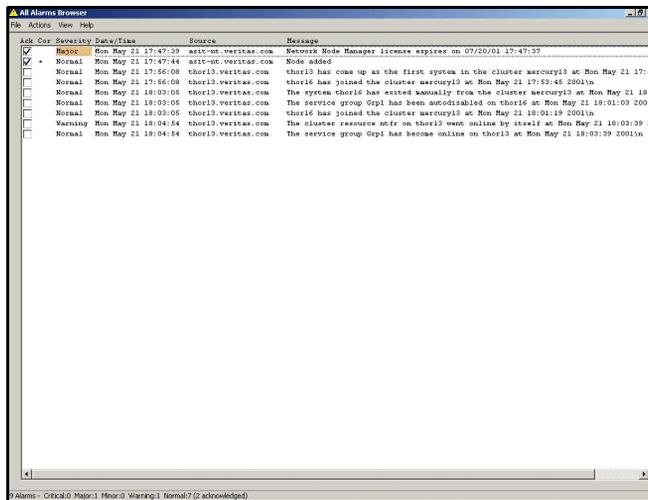
## SNMP-Specific Files

VCS includes two SNMP-specific files: `vcs.mib` and `vcs_trapd`, which are created in `/etc/VRTSvcs/snmp`. The file `vcs.mib` is the textual MIB for built-in traps supported by VCS. Load this MIB into the SNMP console to add it to the list of recognized traps.

The file `vcs_trapd` is specific to the HP OpenView Network Node Manager (NNM) SNMP console, and includes events configured for the built-in SNMP traps supported by VCS. To merge these events with those configured for SNMP traps, type:

```
# xnmevents -merge vcs_trapd
```

When you merge events, the SNMP traps sent by VCS by way of notifier are displayed in HP OpenView NNM SNMP console, as shown below.



**Note** For more information on `xnmevents`, refer to the HP OpenView documentation.

## Configuring Notification

There are two methods for configuring notification: using the ClusterService service group or manually editing the main.cf file as shown in the example configuration file below. For more information on configuring notification, see “NotfierMgr” in the *VERITAS Cluster Server Bundled Agents Reference Guide*.

```
include "types.cf"

cluster VCSCluster142 (
  UserNames = { admin = "lQ+^;2:z" }
)

system north
system south

group NicGrp (
  SystemList = { north, south}
  AutoStartList = { north }
  Parallel = 1
)

Phantom my_phantom (
)

NIC    NicGrp_en0 (
  Enabled = 1
  Device  = en0
  NetworkType = ether
)

group Grp1 (
  SystemList = { north, south }
  AutoStartList = { north }
)

Proxy nicproxy(
  TargetResName = "NicGrp_en0"
)

NotifierMgr ntfr (
  PathName = "/opt/VRTSvc/bin/notifier"
  SnmpConsoles = { snmpserv = Information }
  SmtServer = "smtp.your_company.com"
  SmtRecipients = { "vcsadmin@your_company.com" = SevereError }
)

ntfr requires nicproxy
```



```
// resource dependency tree
//
// group Grp1
// {
//   NotifierMngr ntfr
//   {
//     Proxy nicproxy
//   }
// }
```

## Event Trigger Notification

Event triggers enable the administrator to take specific actions in response to particular events.

### How Event Triggers Work

- ✓ VCS determines if the event is enabled.
- ✓ VCS invokes `hatrigger`.

VCS calls `hatrigger`, a high-level Perl script located at:

```
$VCS_HOME/bin/hatrigger
```

VCS also passes the name of event trigger and the parameters specific to the event. For example, when a service group becomes fully online on a system, VCS invokes `hatrigger -postonline system service_group`. Note that VCS does not wait for `hatrigger` or the event trigger to complete execution. After calling the triggers, VCS continues normal operations.

Event triggers are invoked on the system where the event occurred, with the following exceptions:

- The `SysOffline` and `NoFailover` event triggers are invoked from the lowest-numbered system in `RUNNING` state.
  - The `Violation` event trigger is invoked from all systems on which the service group was brought partially or fully online.
- ✓ The script `hatrigger` invokes an event trigger.

The script `hatrigger` performs actions common to all triggers, and calls the intended event trigger as instructed by VCS. This script also passes the parameters specific to the event.

Event triggers are invoked by event names, for example `violation` denotes a concurrency violation.



## Sample Scripts

VCS provides sample Perl script for each event trigger. These scripts can be customized according to your requirements: you may write your own Perl script. Sample Perl scripts for event triggers are located in `$VCS_HOME/bin/sample_triggers`.

## Event Triggers

The information in the following sections describes the various event triggers, including their usage, parameters, and location.

**Note** Event triggers must reside on all systems in `$VCS_HOME/bin/triggers`. If VCS determines there is no corresponding trigger script or executable in the locations listed for each event trigger, VCS takes no further action.

## InJeopardy Event Trigger

Usage	Location
<p>- <code>injeopardy <i>triggertype</i> <i>system</i> <i>system_state</i></code></p> <p>The variable <code>triggertype</code> represents whether trigger is custom (<code>triggertype=0</code>) or internal (<code>triggertype=1</code>).</p> <p><b>Note</b> For this trigger, <code>triggertype=0</code>.</p> <p>The variable <code>system</code> represents the name of the system.</p> <p>The variable <code>system_state</code> represents the value of the State attribute. See <a href="#">Appendix B</a> for more information on system states.</p>	<p><code>\$VCS_HOME/bin/triggers/InJeopardy</code></p>
<p><b>Description</b></p>	
<p>Invoked when a system is in jeopardy. Specifically, this trigger is invoked when a system has only one remaining link to the cluster, and that link is a network link (LLT). This is a considered a critical event because if the system loses the remaining network link, VCS does not fail over the service groups that were online on the system. Using this trigger to notify the administrator of the critical event enables the administrator to take appropriate action to ensure that the system has at least two links to the cluster.</p> <p>This event trigger is non-configurable.</p>	



## LoadWarning Event Trigger

Usage	Location
<p>- loadwarning <i>triggertype</i> <i>system</i> <i>available_capacity</i></p> <p>The variable <i>triggertype</i> represents whether trigger is custom (<i>triggertype</i>=0) or internal (<i>triggertype</i>=1).</p> <p><b>Note</b> For this trigger, <i>triggertype</i>=0.</p> <p>The variable <i>system</i> represents the name of the system.</p> <p>The variable <i>available_capacity</i> represents the value of the system's AvailableCapacity attribute. (AvailableCapacity=Capacity-sum of Load for system's online groups.)</p> <p>See <a href="#">Appendix B</a> for a complete list of system and group attributes.</p>	<p>\$VCS_HOME/bin/triggers/LoadWarning</p>
<p><b>Description</b></p>	
<p>Invoked when a system becomes overloaded. A system becomes overloaded when the load of the system's online groups exceeds the value designated in the system's LoadWarningLevel attribute (defined as a percentage of the system's capacity in the Capacity attribute) for an interval exceeding the value in the LoadTimeThreshold attribute. For example, say the value of the Capacity attribute is 150, the LoadWarningLevel is 80 and the LoadTimeThreshold is 300. Also, the sum of the Load attribute for all online groups on the system is 135. Because the LoadWarningLevel is 80, safe load is <math>0.80 \times 150 = 120</math>. Actual system load is 135. If system load stays above 120 for more than 300 seconds, the LoadWarningLevel trigger is invoked.</p> <p>Using this trigger to notify the administrator of the critical event enables him or her to switch some service groups to another system, ensuring that no one system is overloaded.</p> <p>This event trigger is non-configurable.</p>	



## NoFailover Event Trigger

Usage	Location
<p data-bbox="105 331 462 388">- nofailover <i>triggertype</i> <i>system</i> <i>service_group</i></p> <p data-bbox="105 413 532 499">The variable <i>triggertype</i> represents whether trigger is custom (<i>triggertype=0</i>) or internal (<i>triggertype=1</i>).</p> <p data-bbox="105 510 476 536"><b>Note</b> For this trigger, <i>triggertype=0</i>.</p> <p data-bbox="105 546 528 633">The variable <i>system</i> represents the name of the last system on which an attempt was made to online the service group.</p> <p data-bbox="105 644 532 701">The variable <i>service_group</i> represents the name of the service group.</p>	<p data-bbox="565 336 991 362">\$VCS_HOME/bin/triggers/NoFailover</p>
<b>Description</b>	
<p data-bbox="105 796 1133 822">Called from the lowest-numbered system in RUNNING state when a service group cannot fail over.</p> <p data-bbox="105 833 508 859">This event trigger is non-configurable.</p>	

## PostOffline Event Trigger

Usage	Location
<p>- postoffline <i>triggertype</i> <i>system</i> <i>service_group</i></p> <p>The variable <i>triggertype</i> represents whether trigger is custom (<i>triggertype=0</i>) or internal (<i>triggertype=1</i>).</p> <p><b>Note</b> For this trigger, <i>triggertype=0</i>.</p> <p>The variable <i>system</i> represents the name of the system.</p> <p>The variable <i>service_group</i> represents the name of the service group that went offline.</p>	<p>\$VCS_HOME/bin/triggers/PostOffline</p>
<p><b>Description</b></p>	
<p>This event trigger is invoked on the system where the group went offline from a partial or fully online state. This trigger is invoked when the group faults, or is taken offline manually.</p> <p>This event trigger is non-configurable.</p>	



## PostOnline Event Trigger

Usage	Location
<p>- <code>postonline <i>triggertype</i> <i>system</i> <i>service_group</i></code></p> <p>The variable <i>triggertype</i> represents whether trigger is custom (<i>triggertype=0</i>) or internal (<i>triggertype=1</i>).</p> <p><b>Note</b> For this trigger, <i>triggertype=0</i>.</p> <p>The variable <i>system</i> represents the name of the system.</p> <p>The variable <i>service_group</i> represents the name of the service group that went online.</p>	SVCS_HOME/bin/triggers/PostOnline
Description	
<p>This event trigger is invoked on the system where the group went online from a partial or fully offline state. This event trigger is non-configurable.</p>	

## ResFault Event Trigger

Usage	Location
<p>- resfault <i>triggertype</i> <i>system</i> <i>resource</i></p> <p>The variable <i>triggertype</i> represents whether trigger is custom (<i>triggertype</i>=0) or internal (<i>triggertype</i>=1).</p> <p><b>Note</b> For this trigger, <i>triggertype</i>=0.</p> <p>The variable <i>system</i> represents the name of the system.</p> <p>The variable <i>resource</i> represents the name of the faulted resource.</p>	<p>\$VCS_HOME/bin/triggers/ResFault</p>
<p><b>Description</b></p>	
<p>Invoked on the system where an online resource has faulted. (The resource has transitioned from ONLINE to FAULTED.) Note that when a resource is faulted, resources within the upward path of the faulted resource are also brought down.</p> <p>This event trigger is non-configurable.</p>	



## ResNotOff Event Trigger

Usage	Location
<p>- resnotoff <i>triggertype system resource</i></p> <p>The variable <i>triggertype</i> represents whether trigger is custom (<i>triggertype=0</i>) or internal (<i>triggertype=1</i>).</p> <p><b>Note</b> For this trigger, <i>triggertype=0</i>.</p> <p>The variable <i>system</i> represents the system on which the resource is not going offline.</p> <p>The variable <i>resource</i> represents the name of the resource.</p>	<p>SVCS_HOME/bin/triggers/ResNotOff</p>
<h3>Description</h3>	
<p>Invoked on the system if a resource in a service group does not go offline even after issuing the offline command to the resource. When invoked, the trigger script waits for a predefined interval and checks the state of the resource. If the resource is not offline, the trigger issues a system shutdown command, followed by the command <code>hastop -local -evacuate</code>.</p> <p>This event trigger is configurable.</p> <p>To configure this trigger, you must define the following attributes:</p> <p><b>Resource Name</b> Define resources for which to invoke this trigger by entering their names in the following line in the script: <code>@resources = ("resource1", "resource2") ;</code></p> <p>If any of these resources do not go offline, the trigger is invoked with that resource name and system name as arguments to the script.</p> <p><b>\$shutdown_timeout</b> Define the time the script waits before it checks the resource state and issues a system shutdown command. For example, if this variable is set to 300, the script waits for 300 seconds before checking that the resource is offline and issuing the shutdown command.</p> <p><b>\$shutdown_countdown</b> Define the time the script waits to shut down the system after issuing the <code>hastop -local -evacuate</code> command. For example, the value 300 indicates that the script waits for 300 seconds after issuing the <code>hastop -local -evacuate</code> command, and then shuts down the system.</p> <p>Define this value to be greater than the time required to switch all service groups on the system to another system.</p> <p><b>\$forced_close_app</b> Define whether the script forcefully closes all running applications when it triggers the system shutdown command. The value 1 indicates the script forcefully closes all running applications. The value 0 indicates it does not. Default is 1.</p> <p><b>\$reboot_option</b> Define whether the script reboots the system after issuing the system shutdown command. The value 1 indicates the script reboots the system. The value 0 indicates it does not. Default is 1.</p>	

## ResStateChange Event Trigger

Usage	Location
<pre data-bbox="149 331 561 387">- resstatechange <i>triggertype</i> system <i>resource</i> <i>previous_state</i> <i>new_state</i></pre> <p data-bbox="149 413 606 499">The variable <i>triggertype</i> represents whether trigger is custom (<i>triggertype</i>=0) or internal (<i>triggertype</i>=1).</p> <p data-bbox="149 508 522 534"><b>Note</b> For this trigger, <i>triggertype</i>=0.</p> <p data-bbox="149 543 635 604">The variable <i>system</i> represents the name of the system.</p> <p data-bbox="149 612 618 673">The variable <i>resource</i> represents the name of the resource.</p> <p data-bbox="149 682 585 743">The variable <i>previous_state</i> represents the resource's previous state.</p> <p data-bbox="149 751 539 812">The variable <i>new_state</i> represents the resource's new state.</p>	<p data-bbox="658 331 1136 357">\$VCS_HOME/bin/triggers/ResStateChange</p>
<h3 data-bbox="149 835 285 861">Description</h3>	
<p data-bbox="149 899 1200 960">This event trigger is not enabled by default. You must enable <code>resstatechange</code> by setting the attribute <code>TriggerResStateChange</code> to 1 in the <code>main.cf</code> file, or by issuing the command:</p> <pre data-bbox="197 977 999 1003"># hagrps -modify <i>service_group</i> TriggerResStateChange 1</pre> <p data-bbox="149 1029 508 1055">This event trigger is configurable.</p> <p data-bbox="149 1081 729 1107">This trigger is invoked under the following conditions:</p> <ul data-bbox="149 1124 1319 1394" style="list-style-type: none"> <li>- Resource goes from OFFLINE to ONLINE.</li> <li>- Resource goes from ONLINE to OFFLINE.</li> <li>- Resource goes from ONLINE to FAULTED.</li> <li>- Resource goes from FAULTED to OFFLINE. (When fault is cleared on non-persistent resource.)</li> <li>- Resource goes from FAULTED to ONLINE. (When faulted persistent resource goes online or faulted non-persistent resource is brought online outside VCS control.)</li> <li>- Resource is restarted by an agent because resource faulted and <code>RestartLimit</code> was greater than 0. (In this case, previous state will be ONLINE and new state will be RESTARTED.)</li> </ul> <p data-bbox="149 1411 1319 1515"><b>Note</b> Use the <code>resstatechange</code> trigger carefully. For example, enabling this trigger for a service group with 100 resources means 100 <code>hatrigger</code> processes and 100 <code>resstatechange</code> processes are fired each time the group is brought online or taken offline. Also, this is not a “wait-mode” trigger. Specifically, VCS invokes the trigger and does not wait for trigger to return to continue operation</p>	



## PreOnline Event Trigger

Usage	Location
<p>- <code>preonline <i>triggertype</i></code>  <code>system <i>service_group whyonlining</i></code></p> <p>The variable <i>triggertype</i> represents whether trigger is custom (<i>triggertype=0</i>) or internal (<i>triggertype=1</i>).</p> <p><b>Note</b> For this trigger, <i>triggertype=0</i>.</p> <p>The variable <i>system</i> represents the name of the system.</p> <p>The variable <i>service_group</i> represents the name of the service group on which the <code>hagrp</code> command was issued or the fault occurred.</p> <p>The variable <i>whyonlining</i> represents two values:</p> <p>FAULT indicates that the group was brought online in response to a group failover or switch.</p> <p>MANUAL indicates that group was brought online manually on the system represented by the variable <i>system</i>.</p>	<p>SVCS_HOME/bin/triggers/PreOnline</p>
<h3>Description</h3>	
<p>Indicates that HAD should not online a service group in response to an <code>hagrp -online</code> command or a fault. It should instead call a user-defined script that checks for external conditions before bringing the group online.</p> <p><b>Note</b> If it is OK to bring the group online, it is then the responsibility of the PreOnline event trigger to bring the group online using the format: <code>hagrp -online -nopre <i>service_group -sys system</i></code></p> <p>If the trigger does not exist, VCS continues to bring the group online.</p> <p>If you do want to bring the group online, define the trigger to take no action.</p> <p>This event trigger is configurable.</p> <ul style="list-style-type: none"> <li>- To enable this trigger, specify <code>PreOnline=1</code> within the group definition, or use:  <code>hagrp -modify <i>service_group</i> PreOnline 1</code></li> <li>- To disable the trigger, specify <code>PreOnline=0</code> within the group definition, or use:  <code>hagrp -modify <i>service_group</i> PreOnline 0</code></li> </ul>	



When you deploy VCS, we recommend that you evaluate performance requirements for:

- ◆ The impact of VCS on overall system performance.
- ◆ The actual VCS performance; for example, the time it takes to failover service groups, etc.

VCS provides various methods that enable you to adjust performance to meet your requirements, including default values that can be used with most applications.

## Impact of VCS on Overall System Performance

VCS and its agents run on the same systems as the applications. Therefore, VCS attempts to minimize its impact on overall system performance. The impact of VCS applies to three main components of clustering: the kernel; specifically, GAB and LLT, the VCS engine (HAD), and the VCS agents. Each is described below. (For details on attributes or commands mentioned in the following sections, see the chapter on administering VCS from the command line and the appendix on VCS attributes.)

### Kernel Components (GAB and LLT)

Typically, overhead of VCS kernel components is minimal. Kernel components provide heartbeat and atomic information exchange among cluster systems. By default, each system in the cluster sends two small heartbeat packets per second to other systems in the cluster. Heartbeat packets are sent over all network links configured in the `/etc/llthosts` configuration file. System-to-system communication also takes place over one network link at a time. **(Ed. note: Please verify. Received comment that preceding sentence incorrect; that is, system-to-system communication takes place over all configured links.)** If a link fails, VCS switches to a different link. Typically, network links are private and do not increase traffic on the public network or LAN. You can configure a public network (LAN) link as low-priority, used only as a heartbeat link, which by default generates one small (approximately 64-byte) broadcast packet per second from each system.



## The VCS Engine “HAD”

The VCS engine, HAD, runs as a daemon process. By default it runs as a high-priority process, which ensures it sends heartbeats to kernel components and responds quickly to failures.

VCS “sits” in a loop waiting for messages from agents, ha commands, the graphical user interfaces, and the other systems. Under normal conditions, the number of messages processed by HAD is few. They mainly include heartbeat messages from agents and update messages from the global counter. VCS may exchange additional messages when an event occurs, but typically overhead is nominal even during events. Note that this depends on the type of event; for example, a resource fault may invoke offlining a group on one system and onlining on another system, but a system fault invokes failing over all of the online service groups on the faulted system.

To continuously monitor VCS status, use the VCS GUI, “Cluster Manager,” or the command `hastatus`. Both methods maintain connection to VCS and register for events, and are more efficient compared to running commands like `hastatus -summary` or `hasys` in a loop.

The number of clients connected to VCS can affect performance if several events occur simultaneously. For example, if five GUI processes are connected to VCS, VCS sends state updates to all five. Maintaining fewer client connections to VCS reduces this overhead.

## The VCS Graphical User Interface

The VCS graphical user interface, “Cluster Manager,” maintains a persistent connection to HAD, from which it receives regular updates regarding cluster status. For best results, run Cluster Manager on a system outside the cluster. This will avoid potential impact on node performance.

## Agents

The VCS agent processes have the most impact on overall system performance. VCS agents handle resource type-specific operations. Agents typically provide `online`, `offline`, `monitor`, and `clean` entry points. Entry points can be implemented in C, C++, or a scripting language such as Perl. There is one agent per resource type.

Each agent process has two components: the agent framework and the agent entry points. The agent framework provides common functionality, such as communication with the HAD, multithreading for multiple resources, scheduling threads, and invoking entry points. Agent entry points implement agent-specific functionality. Follow the performance guidelines below when configuring agents.

### Resource Type and Agent Configuration

By default, VCS monitors each resource every 60 seconds. You can change this by modifying the `MonitorInterval` attribute for the resource type. You may consider reducing monitor frequency for non-critical or resources with expensive monitor operations. Note that reducing monitor frequency also means that VCS may take longer to detect a resource fault.

By default, VCS also monitors offline resources. This ensures that if someone brings the resource online outside of VCS control, VCS detects it and flags a concurrency violation for failover groups. To reduce the monitoring frequency of offline resources, modify the `OfflineMonitorInterval` attribute for the type.

The VCS agent framework uses multithreading to allow multiple resource operations to run in parallel for the same type of resources. For example, a single Mount agent handles all mount resources. Multithreading allows the agent to run the same or different entry points for different resources in parallel. The number of agent threads for most resource types is 10 by default. To change the default, modify the `NumThreads` attribute for the resource type. The maximum value of the `NumThreads` attribute is 20.

The Mount agent schedules the `monitor` entry point for all mount resources, based on the `MonitorInterval` or `OfflineMonitorInterval` attributes. If the number of mount resources is more than `NumThreads`, the monitor operation for some mount resources may be required to wait to execute the `monitor` entry point until the thread becomes free.

Additional considerations for modifying the `NumThreads` attribute include:

- ◆ If you have only one or two resources of a given type, you can set `NumThreads` to a lower value.
- ◆ If you have many resources of a given type, evaluate the time it takes for the `monitor` entry point to execute and the available CPU power for monitoring. For example, if you have 50 mount points, you may want to increase `NumThreads` to get the ideal performance for the Mount agent without affecting overall system performance.



You can also adjust how often VCS monitors various entry points by modifying their associated attributes. The attributes `MonitorTimeout`, `OnlineTimeOut`, and `OfflineTimeout` indicate the maximum time (in seconds) within which the `monitor`, `online`, and `offline` entry points must complete or else be terminated. The default for the `MonitorTimeout` attribute is 60 seconds. The defaults for the `OnlineTimeout` and `OfflineTimeout` attributes is 300 seconds. For best results, we recommend measuring the length of time it takes to bring a resource online, take it offline, and monitor before modifying the defaults. Simply issue an `online` or `offline` command to measure the time it takes for each action. To measure how long it takes to monitor a resource, fault the resource then issue a `probe`, or bring the resource online outside of VCS control and issue a `probe`.

Agents typically run with normal priority. When you develop agents, consider the following:

- ◆ If you write a custom agent, write the `monitor` entry point using C or C++. If you write a script-based monitor, VCS must invoke a new process each time with the `monitor`. This can be costly if you have many resources of that type.
- ◆ If monitoring the resources type is proving costly, you can divide it into “shallow” and “deep” monitoring. Shallow monitoring checks only for the existence of the process. Deep monitoring performs detailed checking of the resource; for example, deep monitoring a database service can send an SQL query. Whether to use shallow or deep monitoring depends on your configuration requirements. You may decide to perform one deep monitoring after every three shallow, or decide to make it configurable.

### Additional Considerations for Agents

Properly configure the attribute `SystemList` for your service group. For example, if you know that a service group can go online on `sysa` and `sysb` only, *do not* include other systems in the `SystemList`. This saves additional agent processes and monitoring overhead.

## VCS Performance

This section describes factors that affect VCS operations, such as bringing a resource or service group online, taking them offline, and failing service groups over to a different system.

### Booting a Cluster System

When a cluster system boots, the kernel drivers and VCS process start in a particular order. If it is the first system in the cluster, VCS reads the cluster configuration file `main.cf` and builds an “in-memory” configuration database. This is the `LOCAL_BUILD` state. When the system finishes building the configuration database, it transitions into the `RUNNING` mode. If another system joins the cluster while the first system is in the `LOCAL_BUILD` state, it must wait until the first system transitions into `RUNNING` mode. The time it takes to build the configuration depends on the number of service groups in the configuration and their dependencies, and the number of resources per group and resource dependencies. VCS creates an object for each system, service group, type, and resource. Typically, the number of systems, service groups and types are few, so the number of resources and resource dependencies determine how long it takes to build the configuration database and get VCS into `RUNNING` mode. If a system joins a cluster in which at least one system is in `RUNNING` mode, it builds the configuration from the lowest-numbered system in that mode.

---

**Note** Onlining service groups as part of AutoStart occurs after VCS transitions to `RUNNING` mode.

---

### Bringing a Resource Online

The `online` entry point of an agent brings the resource online. This entry point may return before the resource is fully online. The subsequent monitor determines if the resource is online, then reports that information to VCS. The time it takes to bring a resource online equals the time for the resource to go online, plus the time for the subsequent monitor to execute and report to VCS.

Most resources are online when the `online` entry point finishes. The agent schedules the monitor immediately after the entry point finishes, so the first monitor detects the resource as online. However, for some resources, such as a database server, recovery can take longer. In this case, the time it takes to bring a resource online depends on the amount of data to recover. It may take multiple monitor intervals before a database server is reported online. When this occurs, it is important to have the correct values configured for the `OnlineTimeout` and `OnlineWaitLimit` attributes of the database server resource type.



## Taking a Resource Offline

Similar to the `online` entry point, the `offline` entry point takes the resource offline and may return before the resource is actually offline. Subsequent monitoring confirms whether the resource is offline. The time it takes to offline a resource equals the time it takes for the resource to go offline, plus the duration of subsequent monitoring and reporting to VCS that the resource is offline. Most resources are typically offline when the offline entry point finishes. The agent schedules the monitor immediately after the `offline` entry point finishes, so the first monitor detects the resource as offline.

## Bringing a Service Group Online

The time it takes to bring a service group online depends on the number of resources in the service group, the service group dependency structure, and the time to bring the group's resources online. For example, if service group G1 has three resources, R1, R2, and R3 (where R1 depends on R2 and R2 depends on R3), VCS first onlines R3. When R3 is online, VCS onlines R2. When R2 is online, VCS onlines R1. The time it takes to online G1 equals the time it takes to bring all resources online. However, if R1 depends on both R2 and R3, but there was no dependency between them, the online operation of R2 and R3 is started in parallel. When both are online, R1 is brought online. The time it takes to online the group is Max (the time to online R2 and R3), plus the time to online R1. Typically, broader service group trees allow more parallel operations and can be brought online faster. Deeper service group trees do not allow much parallelism and serializes the group online operation.

## Taking a Service Group Offline

Service group offlining works from the top down, as opposed to onlining, which works from the bottom up. The time it takes to offline a service group depends on the number of resources in the service group and the time to offline the group's resources. For example, if service group G1 has three resources, R1, R2, and R3, VCS first offlines R1. When R1 is offline, VCS offlines R2. When R2 is offline, VCS offlines R3. The time it takes to offline G1 equals the time it takes for all resources to go offline.

## Detecting Resource Failure

The time it takes to detect a resource fault or failure depends on the `MonitorInterval` attribute for the resource type. When a resource faults, the next monitor detects it. The agent may not declare the resource as faulted if the `ToleranceLimit` attribute is set to non-zero. If the `monitor` entry point reports offline more often than the number set in `ToleranceLimit`, the resource is declared faulted. However, if the resource remains online for the interval designated in `ConfInterval`, any earlier reports of offline are not counted against `ToleranceLimit`.

When the agent determines that the resource is faulted, it calls the `clean` entry point, if implemented. This is done to verify that the resource is completely offline. The next monitor after `clean` confirms the offline. The agent then tries to online the resource again if `RestartLimit` is non-zero. The agent attempts to restart the resource according to the number set in `RestartLimit` before it gives up and informs HAD that the resource is faulted. However, if the resource remains online for the interval designated in `ConfInterval`, earlier attempts to restart are not counted against `RestartLimit`.

In most cases, `ToleranceLimit` is 0. The time it takes to detect a resource failure is the time it takes the agent monitor to detect failure, plus the time to clean up the resource if the `clean` entry point is implemented. Therefore, the time it takes to detect failure depends on the `MonitorInterval`, the efficiency of the `monitor` and `clean` (if implemented) entry points, and the `ToleranceLimit` (if set).

In some cases, the failed resource may hang and may also cause the monitor to hang. For example, if the database server is hung and the monitor tries to query, the monitor will also hang. If the `monitor` entry point is hung, the agent eventually kills the thread running the entry point. By default, the agent times out the `monitor` entry point after 60 seconds. This can be adjusted by changing the `MonitorTimeout` attribute. The agent retries `monitor` after the `MonitorInterval`. If the `monitor` entry point times out consecutively for the number of times designated in the attribute `FaultOnMonitorTimeouts`, the agent treats the resource as faulted. The agent calls `clean`, if implemented. The default value of `FaultOnMonitorTimeouts` is 4, and can be changed according to the type. A high value of this parameter delays detection of a fault if the resource is hung. If the resource is hung and causes the `monitor` entry point to hang, the time to detect it depends on `MonitorTimeout`, `FaultOnMonitorTimeouts`, and the efficiency of `monitor` and `clean` (if implemented).



## Detecting System Failure

When a system crashes or is powered off, it stops sending heartbeats to other systems in the cluster. By default, other systems in the cluster wait 21 seconds before declaring it dead. The time of 21 seconds derives from 16 seconds default timeout value for LLT peer inactive timeout, plus 5 seconds default value for GAB stable timeout. The default peer inactive timeout is 16 seconds, and can be modified in the `/etc/llttab` file. For example, to specify 12 seconds, type:

```
set-timer peerinact:1200
```

---

**Note** After modifying the peer inactive timeout, you must unconfigure, then restart LLT before the change is implemented. To unconfigure LLT, type `lltconfig -u`. To restart LLT, type `lltconfig -c`.

---

GAB stable timeout can be changed by specifying:

```
gabconfig -t timeout_value_milliseconds
```

Though this can be done, we *do not* recommend changing the values of the LLT peer inactive timeout and GAB stable timeout.

If a system reboots, it becomes unavailable until the reboot is complete. The reboot process kills all processes, including HAD. When the VCS process is killed, other systems in the cluster mark all service groups that can go online on the rebooted system as autodisabled. The AutoDisabled flag is cleared when the system goes offline. As long as the system goes offline within the interval specified in the ShutdownTimeout value, VCS treats this as a system reboot. The ShutdownTimeout default value of 120 can be changed by modifying the attribute. See “[System Attributes](#)” on page 353 for details.

## Detecting Network Link Failure

If a system loses a network link to the cluster, other systems stop receiving heartbeats over the links from that system. As mentioned above, LLT detects this and waits for 16 seconds before declaring the system lost a link.

## When a System Panics

There are several instances in which GAB will intentionally panic a system, including if it detects an internal protocol error or discovers an LLT node-ID conflict. Three other instances are described below.

### Client Process Failure

If the client process fails to heartbeat to GAB, the process is killed. If the process hangs in the kernel and cannot be killed, GAB halts the system. If the `-k` option is used in the `gabconfig` command, GAB tries to kill the client process until successful, which may have an impact on the entire cluster. If the `-b` option is used in `gabconfig`, GAB does not try to kill the client process. Instead, the system is halted when client process fails to heartbeat.

### Network Failure

If a network partition occurs, a cluster can “split” into two or more separate mini-clusters. When two clusters join as one, VCS designates that one system be ejected. GAB prints diagnostic messages and sends `iofence` messages to the system being ejected. The system receiving the `iofence` messages tries to kill the client process. The `-k` option applied here. If the `-j` option is used in `gabconfig`, the system is halted when the `iofence` message is received. See [Chapter 13](#) for more information on network failure, partitioning, and split brain.

### Quick Reopen

If a system leaves cluster and tries to join the cluster before the new cluster is configured (default is five seconds), the system is sent an `iofence` message with reason set to “quick reopen.” When the system receives the message, it tries to kill the client process.



## Service Group Switch

The time it takes to switch a service group equals the time to offline a service group on the source system, plus the time to bring the service group online on the target system.

## Service Group Failover

The time it takes to fail over a service group when a resource faults equals

- ◆ the time it takes to detect the resource fault
- ◆ the time it takes to offline the service group on source system
- ◆ the time it takes for the VCS policy module to select target system
- ◆ the time it takes to bring the service group online on target system

The time it takes to failover a service group when a system faults equals

- ◆ the time it takes to detect system fault
- ◆ the time it takes to offline the service group on source system
- ◆ the time it takes for the VCS policy module to select target system
- ◆ the time it takes to bring the service group online on target system

The time it takes the VCS policy module to determine the target system is negligible in comparison to the other factors.

If you have a firm group dependency and the child group faults, VCS offlines all immediate and non-immediate parent groups before bringing the child group online on the target system. Therefore, the time it takes a parent group to be brought online also depends on the time it takes the child group to be brought online.

## Scheduling Class and Priority Configuration Support

VCS allows you to specify priorities and scheduling classes for VCS processes. VCS supports the following scheduling classes:

- ◆ RealTime (specified as “RT” in the configuration file)
- ◆ TimeSharing (specified as “TS” in the configuration file)

### Priority Ranges

The following table displays the platform-specific priority range for RealTime, TimeSharing, and SRM scheduling (SHR) processes.

Platform	Scheduling Class	Default Priority Range Weak / Strong	Priority Range Using #ps Commands
AIX	RT TS	126 / 50 60	126 / 50 Priority varies with CPU consumption. <b>Note</b> On AIX, use #ps -ae1
HP-UX	RT TS	127 / 0 N/A	127 / 0 N/A <b>Note</b> On HP-UX, use #ps -ae1
Linux	RT TS	1 / 99	L-high priority task N-high priority task <b>Note</b> On Linux, use #ps -ae1
Solaris	RT TS SHR	0 / 59 -60 / 60 -60 / 60	100 / 159 N/A N/A <b>Note</b> On Solaris, use #ps -ae -o pri, args



## Default Scheduling Classes and Priorities

The following table lists the default class and priority values used by VCS. Note that the default priority value is platform-specific. Therefore, when priority is set to "" (empty string), VCS converts the priority to a value specific to the platform on which the system is running. For TS, the default priority equals the strongest priority supported by the TimeSharing class. For RT, the default priority equals two less than the strongest priority supported by the RealTime class. So, if the strongest priority supported by the RealTime class is 59, the default priority for the RT class is 57. For SHR (on Solaris only), the default priority is the strongest priority supported by the SHR class.

The class and priority of trigger processes are determined based on the attributes ProcessClass (default = TS) and ProcessPriority (default = ""). Both attributes can be modified according to the class and priority at which the trigger processes run.

Process	Default Scheduling Class	Default Priority			
		AIX	HP-UX	Linux	Solaris
Engine	RT	52 (Strongest + 2)	2 (Strongest + 2)	Min: 0 Max: 99	57 (Strongest - 2)
Process created by engine	TS	60	N/A	Always 0	60 (Strongest)
Agent	TS	60	N/A	Always 0	60 (Strongest)
Script	TS	60	N/A	Always 0	60 (Strongest)

**Note** For standard configurations, we recommend using the default values for scheduling unless specific configuration requirements dictate otherwise.



## VCS Logging

VCS generates several types of messages, three of which (Tags A–C) indicate actual errors:

- ◆ TAG\_A indicates a critical error within a VCS process. Contact Technical Support immediately.
- ◆ TAG\_B indicates failure of a cluster component, unanticipated state change, or termination or unsuccessful completion of a VCS action.
- ◆ TAG\_C indicates a warning or error, but not an actual fault.
- ◆ TAG\_D informs the user that VCS has initiated an action.
- ◆ TAG\_E informs the user of various state messages or comments.

VCS generates two error message logs: the engine log and the agent log. Log file names are appended by letters. Letter “A” indicates the first log file, “B” the second, “C” the third, and so on.

The engine log is located at `/var/VRTSvcs/log/engine_A.log`. The format of engine log messages is:

```
TAG | Date (yy/mm/dd) | Time | Product | Message ID | Message
```

A typical engine log resembles:

```
TAG_C 2001/12/03 18:54:58 VCS:10075:building from remote system.
```

The agent log is located at `$VCS_HOME/log/agent_A.log`. The format of agent log messages is:

```
TAG | Date (yy/mm/dd) | Time | Product | Message ID | Agent Type |  
Resource Name | Entry Point | Message
```

A typical agent log resembles:

```
TAG_C 2001/12/03 19:40:01 VCS:254001:Oracle:oracle1:online:  
One or more attributes contain invalid values.
```

TAG A–E messages are logged to the engine and the agent logs. Tag A–B messages are also written to the system log. You can view these message logs using any text editor.



## Message Catalogs

VCS includes multilingual support for message catalogs. These binary message catalogs (BMCs), are stored in the following default locations. The variable *language* represents a two-letter abbreviation.

```
/opt/VRTSvcs/messages/language/module_name  
/opt/VRTSgab/messages/language/module_name  
/opt/VRTSllt/messages/language/module_name
```

The VCS command-line interface displays error/success messages in any language supported by VCS. Additionally, a new command, `hamsg`, displays the VCS engine logs in VCS-supported languages.

The following table shows the complete list of BMCs.

Module Name	Description
<code>VRTSvcsHad.bmc</code>	VCS engine (HAD) messages
<code>VRTSvcsAgentplatform.bmc</code>	VCS bundled agent messages
<code>VRTSvcsplatformagent_name.bmc</code>	VCS enterprise agent messages
<code>gab.bmc</code>	GAB command-line interface messages
<code>llt.bmc</code>	LLt command-line interface messages

## Network Partitions and Split-Brain

Under normal conditions, when a VCS system ceases heartbeat communication with its peers due an event such as power loss or a system crash, the peers assume the system has failed and issue a new, “regular” membership excluding the departed system. A designated system in the cluster then takes over the service groups running on the departed system, ensuring the application remains highly available.

However, heartbeats can also fail due to network failures. If all network connections between any two groups of systems fail simultaneously, a *network partition* occurs. When this happens, systems on both sides of the partition can restart applications from the other side resulting in duplicate services, or “split-brain.” A split brain occurs when two independent systems configured in a cluster assume they have exclusive access to a given resource (usually a file system or volume). The most serious problem caused by a network partition is that it affects the data on shared disks.

All failover management software uses a predefined method to determine if its peer is “alive.” If the peer is alive, the system recognizes it cannot safely take over resources. Split brain occurs when the method of determining peer failure is compromised. In virtually all FMS systems, split-brain situations are rare. A true split brain means multiple systems are online and have accessed an exclusive resource simultaneously.

---

**Note** *Splitting communications between cluster nodes does not constitute a split brain. A split-brain means cluster membership was affected in such a way that multiple systems use the same exclusive resources, usually resulting in data corruption.*

---

The goal is to minimize the chance of a system taking over an exclusive resource while another has it active, yet accommodate a system powering off. In other words, a way to discriminate between a system that has failed and one that is simply not communicating.



## How VCS Avoids Split Brain

VCS uses heartbeats to determine the “health” of its peers. These can be private network heartbeats, public (low-priority) heartbeats, and disk heartbeats. Regardless of the heartbeat configuration, VCS determines that a system has faulted (due to power loss, kernel panic, etc.) when *all* heartbeats fail simultaneously. For this method to work, the system must have two or more functioning heartbeats and all must fail simultaneously.

For VCS to encounter split brain, the following events must occur:

- ◆ A service group must be online on a system in a cluster.
- ◆ The service group must have a system (or systems) designated in its SystemList attribute as a potential failover target.
- ◆ All heartbeat communication between the system with the online service group and the system designated as the potential takeover target must fail simultaneously while the original system stays online.
- ◆ The potential takeover target must actually bring resources online that are typically an exclusive, ownership-type item, such as disk groups, volume, or file systems.

## Jeopardy Defined

The design of VCS requires that a minimum of two heartbeat-capable channels be available between nodes to protect against network failure. When a node is missing a single heartbeat connection, VCS can no longer discriminate between a system loss and a loss of the last network connection. It must then handle loss of communications on a single network differently from loss on multiple networks. This procedure is called “jeopardy.”

As mentioned previously, LLT provides notification of reliable versus unreliable network communications to GAB. GAB uses this information, with or without a functional disk heartbeat, to delegate cluster membership. If a system’s heartbeats are lost simultaneously across all channels, VCS determines the system has failed. The services running on that system are then restarted on another. However, if the node was running with one heartbeat only (in jeopardy) prior to the loss of a heartbeat, VCS does *not* restart the applications on a new node. This action of disabling failover is a safety mechanism that prevents data corruption.

A system can be placed in a jeopardy membership under two conditions:

- ◆ The system has only one functional network heartbeat and no disk heartbeat.

In this situation, the node is a member of the regular membership and the jeopardy membership. Being in a regular membership and jeopardy membership simultaneously changes the failover on system-fault behavior only. All other cluster functions remain the same. This means failover due to a resource fault, or switchover of service groups at operator request, are unaffected. The only change is disabling other systems from assuming service groups on system fault. VCS continues to operate as a single cluster when at least one network channel exists between the systems. However, when only one channel remains, failover due to system failure is disabled. Even after the last network connection is lost, VCS continues to operate as partitioned clusters on each side of the failure.

- ◆ The system has no network heartbeat and only a disk heartbeat.

As mentioned in “[Cluster Communications](#)” on page 18, disk heartbeats are incapable of carrying cluster status. In this scenario, the node is excluded from the regular membership because it is impossible to track status of resources on the node, and it is placed in a special jeopardy membership only. Failover on resource fault or operator-initiated switchover is disabled. VCS prevents any actions on service groups running on the departed system because it is impossible to ascertain from a disk heartbeat the status of resources on the system. Reconnecting the network without stopping VCS and GAB results in one or more systems halting, or one or more systems stopping and restarting HAD and associated service groups.



## Reconnecting the Private Network

As mentioned in the preceding section, when a final network connection is lost, the systems on each side of the network partition do not stop, they segregate into “mini-clusters.” Each cluster continues to operate and provide services; however, failover of any service group to or from the opposite side of the partition is disabled. This design enables administrative services to operate uninterrupted; for example, you can use VCS to shut down applications during system maintenance. When the cluster is split, reconnecting the private network must be undertaken with care.

Reconnecting a private network after a cluster has been segregated no longer causes system panic. Instead, HAD stops and restarts (“recycles”), which halts all HA services.

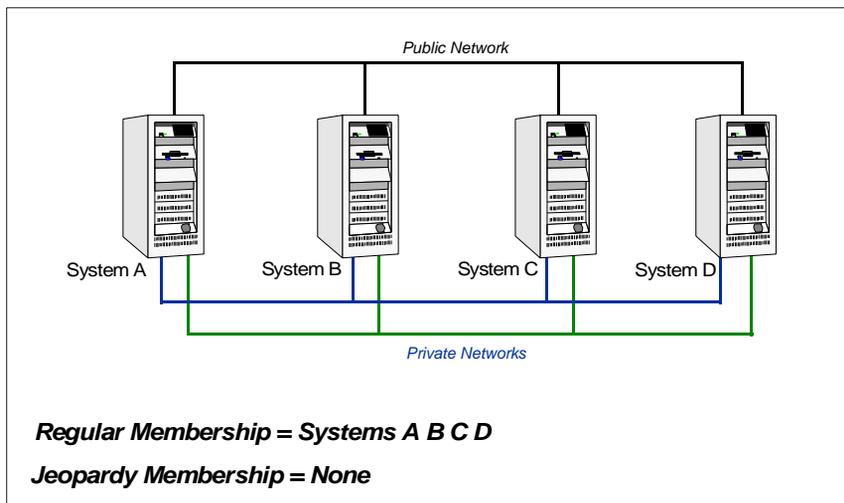
There are several rules that determine which systems are affected.

- ◆ On a two-node cluster, the system with the lowest LLT host ID stays running and the system with the higher ID recycles HAD.
- ◆ In a multinode cluster, the largest running group stays running. The smaller groups recycle HAD.
- ◆ On a multinode cluster that splits into two, equal-sized clusters, the cluster with the lowest node number stays running. The higher group recycles HAD.

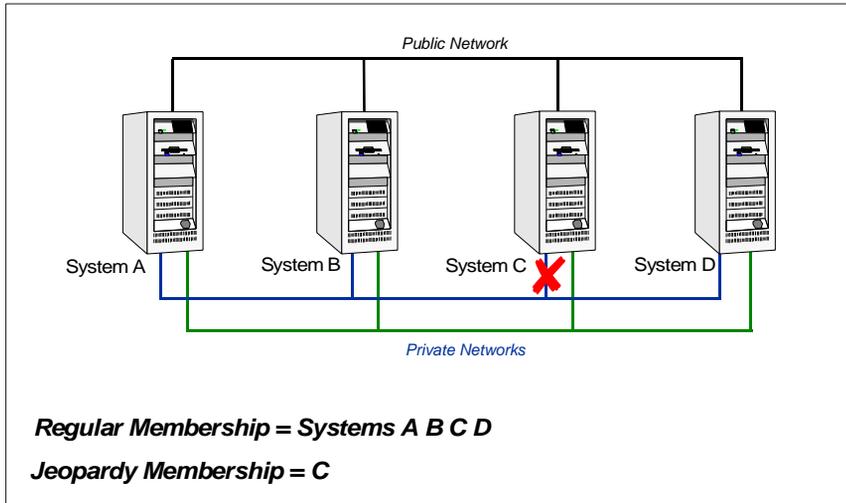
## Examples of Jeopardy and Network Partitions

### Example 1: Four-Node Cluster with Two Private Heartbeat Connections

This configuration has no low-priority link and no disk heartbeat. In a standard configuration, both private links are load-balancing cluster status and both carry heartbeats. The figure below shows the basic VCS communications configuration.

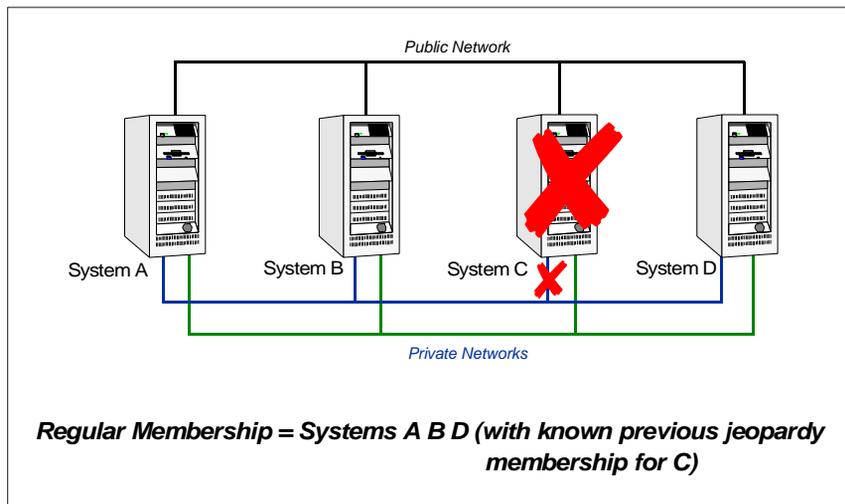


If a link to System C fails, the system is rendered in an unreliable communications state because there is only one heartbeat. A new cluster membership is issued with Systems A, B, C, and D in the regular membership and System C in a jeopardy membership. All normal cluster operations continue, including normal failover of service groups due to resource fault.



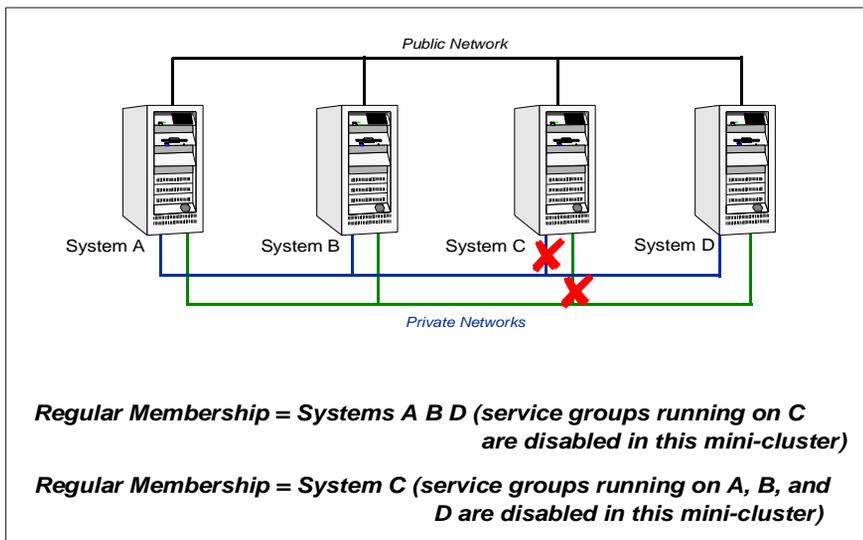
Now if System C fails due to loss of power, the other systems in the cluster recognize it has faulted. In this situation, a new membership is issued for Systems A, B, and D as regular members and no jeopardy membership. No further action is required.

Because System C was in a jeopardy membership, any service group on the system is AutoDisabled, meaning no other system will attempt to assume ownership of the groups. If the System C is actually failed, the system administrator can clear the AutoDisabled flag on the service groups and bring them online on other systems in the cluster. Clearing the AutoDisabled flag informs VCS that the system is actually down.



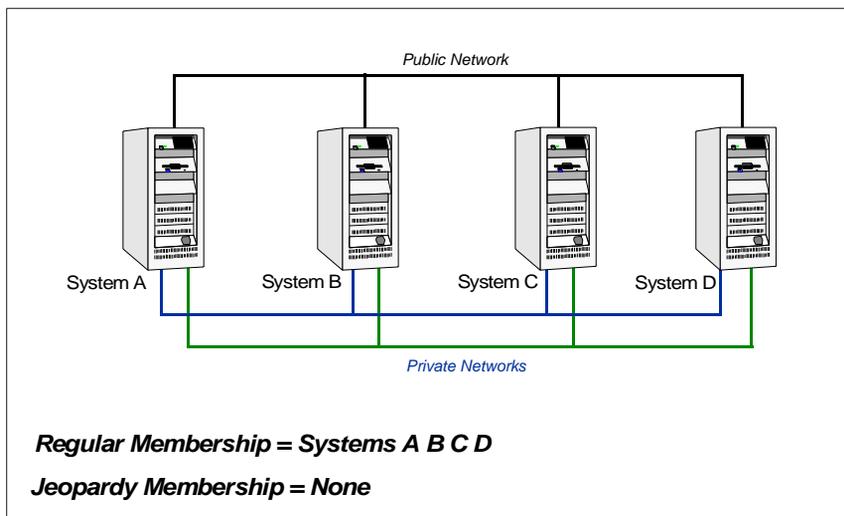
In the final scenario of this example, the configuration is reset and System C is operating in the cluster with one heartbeat. If this heartbeat to System C is lost, a new membership is issued for Systems A, B, and D as regular members and no jeopardy membership. Because System C was in a jeopardy membership, any service group running on the system is AutoDisabled to prevent other systems from assuming ownership of the groups.

Systems A, B, and D become a mini-cluster of three nodes, and System C becomes its own mini-cluster. All service groups on Systems A, B, and D are autodisabled to System C due to the previous jeopardy membership. System C issues its own new membership with itself as the sole, regular member. Reconnecting either LLT link causes HAD to recycle on System C.

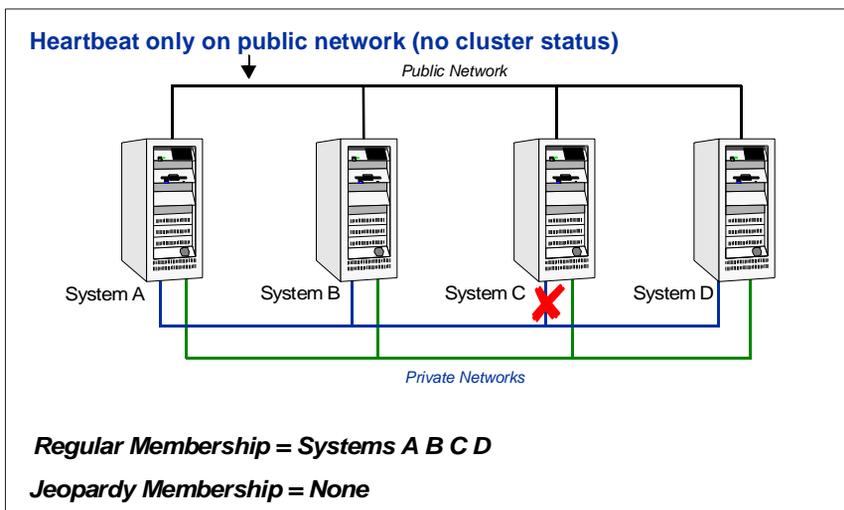


### Example 2: Four-Node Cluster with Two Private Networks and One Low-Priority, Public Network

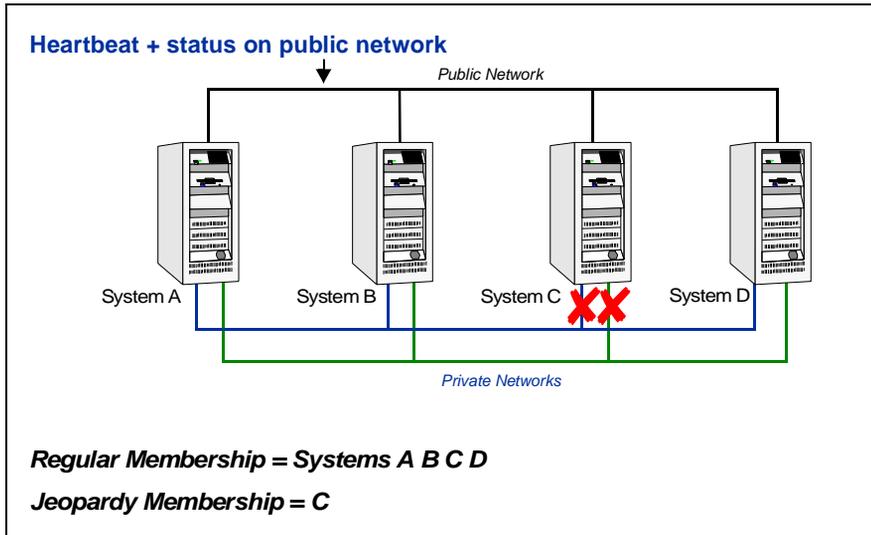
In this configuration, cluster status is load-balanced across the two private links and heartbeats are sent on all three links. The public network heartbeat is reduced in frequency to twice per second.



If a private link to System C is lost, the other nodes send all cluster status traffic to System C via the remaining private link and use both private links for traffic between themselves. The low-priority link continues with heartbeat only. No jeopardy condition exists because there are two links to detect system failure.



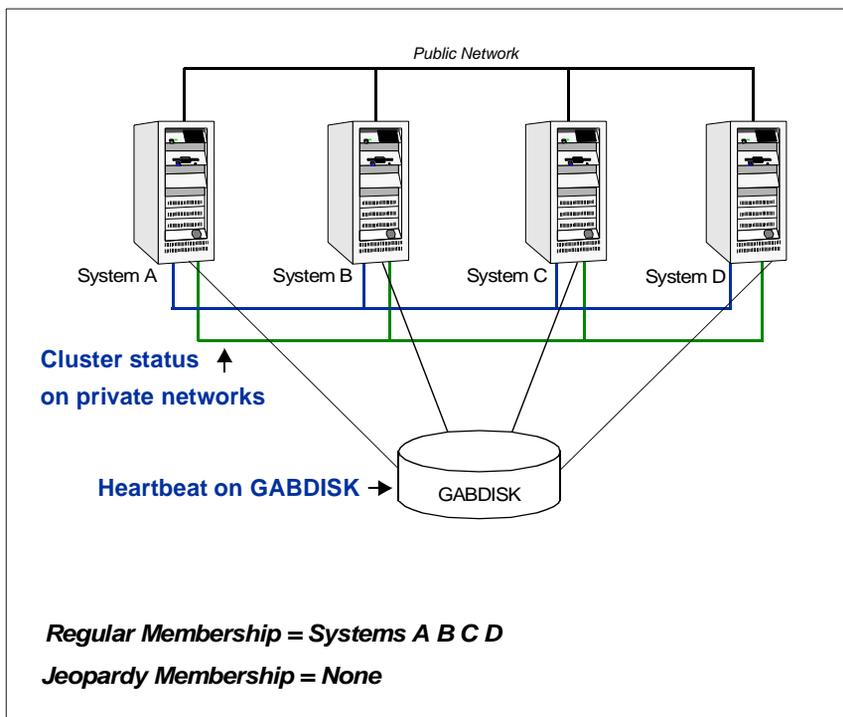
If the second private heartbeat link is lost, cluster status communication is routed over the public network link to System C. System C is then placed in a jeopardy membership and autofailover on the system is disabled.



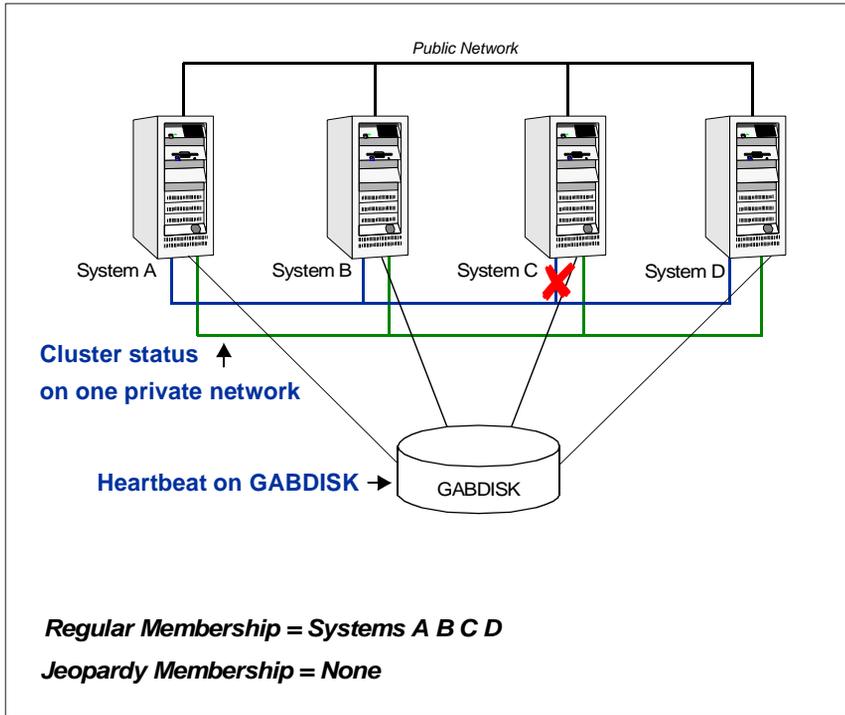
When the private network is reconnected, cluster status reverts to the private link and the low-priority link returns to heartbeat only. System C is returned to normal regular membership with no jeopardy membership.

**Example 3: Four-Node Cluster with Two Private Heartbeats and One Disk Heartbeat**

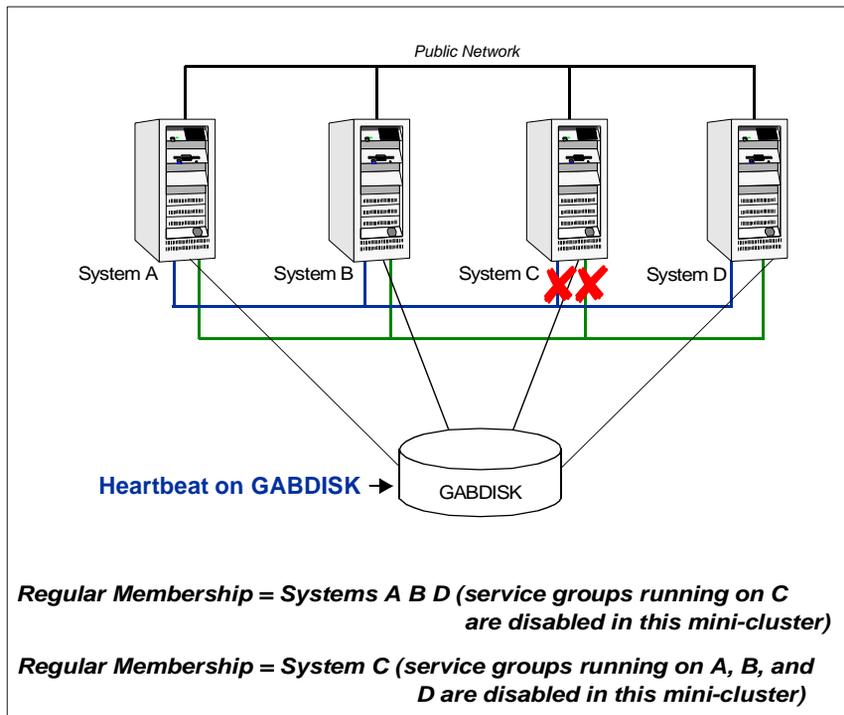
Under normal conditions, all cluster status is load-balanced across the two private networks. Heartbeats are sent on both network channels. GABDISK (or gabdiskhb) places another heartbeat on the disk.



If a private heartbeat link is lost, all cluster status shifts to the remaining private link. There is no jeopardy at this point because two heartbeats remain available to detect system failure.



If the second heartbeat is lost, no cluster status channel is available so the cluster splits into mini-clusters. Because heartbeats continue to write to disk, systems on each side of the “split” autotisable service groups running on the opposite side. Note that for this type of jeopardy membership there is no corresponding regular membership.



Reconnecting one of the private links causes HAD to recycle.



## Preexisting Network Partitions

A *preexisting network partition* refers to failures in communication channels that occur while the systems are down. Regardless of whether the cause is scheduled maintenance or system failure, VCS cannot respond to failures when systems are down. This leaves VCS vulnerable to network partitioning when the systems are booted.

A key concept for understanding VCS is “probing.” During startup, VCS performs a monitor sequence, or “probe,” on all resources configured in the cluster to ascertain which resources are online on any system. This prevents potential concurrency violations caused by a system administrator starting resources manually, outside VCS control.

VCS can only communicate with nodes that are part of the LLT network. For example, consider a four-node cluster (Systems A, B, C, and D) that is shutdown for weekend maintenance. While the cluster is down, the heartbeat connections are severed to System D. A system administrator is then directed to bring the Oracle database back up. If he or she manually brings up Oracle on System D, he or she will likely encounter a problem. If VCS were allowed to start on Systems A–C, the systems would not recognize System D and its online resources, resulting in split-brain. VCS seeding prevents such situations, as described below.

### 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.

Seeding control is established in the `/etc/gabtab` file. GAB is started with the command line `/sbin/gabconfig -c -n X` where *X* is equal to the total number of nodes in the cluster. For example, the command line in the `/etc/gabtab` file for a four-node cluster is `/sbin/gabconfig -c -n 4`. VCS requires you declare the number of systems participating 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.

To start the cluster with fewer nodes than the total, verify the nodes to be excluded are actually down, then start GAB with the command `/sbin/gabconfig -c -x`. This manually seeds the cluster and enables VCS to start on all connected systems.

During initial startup, VCS autodisables a service group until all of the group's resources are probed on systems in the SystemList that have GAB running. This protects against a situation where enough systems are running LLT and GAB to seed the cluster, but not all of them have HAD running.

Note that before VCS can accept HA commands, the cluster nodes must be seeded. If the nodes are not seeded and you attempt to issue a command, you receive the error message:

```
VCS:11037:Node has not received cluster membership yet,  
cannot process HA command
```

## Preonline IP Check

You can now enable a preonline check of a failover IP address to protect against network partitioning. The check pings a service group's configured IP address to verify it is not already in use. If it is, the service group is not brought online. A second check verifies the system is connected to its public and private networks. If the system receives no response from a broadcast ping to the public network and a check of the private networks, it determines the system is isolated and does not bring the service group online.

### ▼ To enable the preonline IP check

1. Move the preonline trigger script from the sample triggers directory into the triggers directory:

```
# cp /opt/VRTSvcs/bin/sample_triggers/preonline_ipc  
/opt/VRTSvcs/bin/triggers/preonline
```

2. Change the file permissions to make it executable.

## When VCS Shuts Down a System

In some cases, VCS kernel components may intentionally bring down a system to avoid network partitioning. See the *VERITAS Cluster Server Release Notes* for details.



## Network Partitions and the UNIX Boot Monitor

Most UNIX systems provide a console-abort sequence that enables you to halt and continue the processor. On AIX systems, this is the Ctrl -| keyboard sequence, which puts the system in the kernel debugger mode, provided the mode (kdb/lldb) is enabled.

*Continuing operations after the processor has stopped may corrupt data and is therefore unsupported by VCS.* 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 another console sequence, it continues writing to shared storage as before, even though its applications have been restarted on other systems where available.

If a write operation was pending when the console-abort sequence was processed, the write occurs immediately after the `quit` command is typed from the debugger prompt. (Typing `quit` from the debugger prompt exits the debugger.) Halting a system by this method appears to all other nodes as a complete system fault because all heartbeats disappear simultaneously. Another node takes over services for the missing node. When the resume occurs, it takes several seconds before the return of a formerly missing heartbeat causes a system panic. During this time, the write waiting on the stopped node occurs, leading to data corruption.

We recommend rebooting the system if it was halted using the console-abort sequence. (Typing `reboot` from the debugger prompt reboots the system.)



## Disabling Resources

Disabling a resource means that the resource is no longer monitored by a VCS agent, and that the resource cannot be brought online or taken offline. The agent starts monitoring the resource after the resource is enabled. The resource attribute `Enabled` determines whether a resource is enabled or disabled. (See “[Resource Attributes](#)” on page 334 for details.) A persistent resource can be disabled when all its parents are offline. A non-persistent resource can be disabled when the resource is in an `OFFLINE` state.

### When to Disable a Resource

Typically, resources are disabled when one or more resources in the service group encounter problems and disabling the resource is required to keep the service group online or to bring it online.

---

**Note** Disabling a resource is not an option when the entire service group requires disabling. In that case, set the service group attribute `Enabled` to 0.

---

#### ▼ To disable a resource

To disable the resource when VCS is running, type:

```
# hares -modify resource Enabled 0
```

To have the resource disabled initially when VCS is started, set the resource's `Enabled` attribute to 0 in `main.cf`.

### Limitations

When VCS is running, there are certain prerequisites to be met before the resource is disabled successfully.

- ✓ An online non-persistent resource cannot be disabled. It must be in a clean `OFFLINE` state. (The state must be `OFFLINE` and `IState` must be `NOT WAITING`.)
- ✓ If it is a persistent resource and the state is `ONLINE` on some of the systems, all dependent resources (parents) must be in clean `OFFLINE` state. (The state must be `OFFLINE` and `IState` must be `NOT WAITING`.)

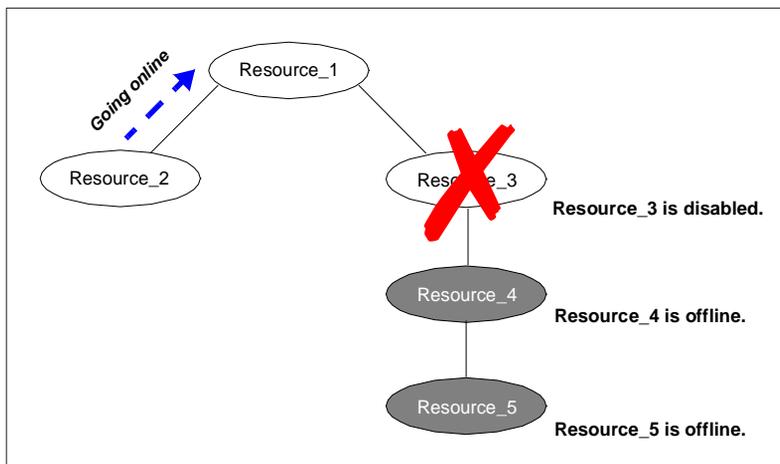
Therefore, before disabling the resource you may be required to take it offline (if it is non-persistent) and take other resources offline in the service group.



## Additional Considerations

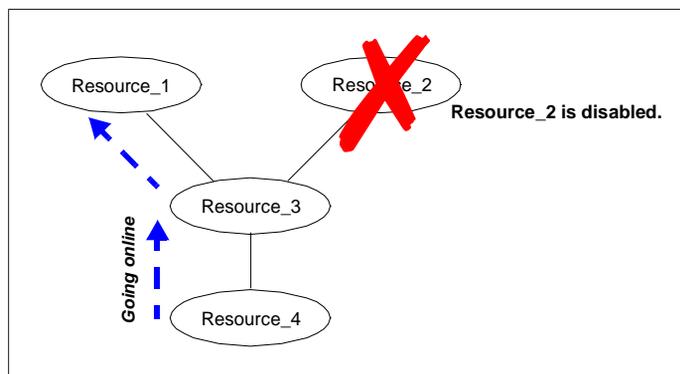
- ◆ When a group containing disabled resources is brought online, the online transaction is not propagated to the disabled resources. Children of the disabled resource are brought online by VCS only if they are required by another enabled resource.
- ◆ You can bring children of disabled resources online if necessary.
- ◆ When a group containing disabled resources is taken offline, the offline transaction is propagated to the disabled resources.

The following figures show how a service group containing disabled resources is brought online.



In the figure above, Resource\_3 is disabled. When the service group is brought online, the only resources brought online by VCS are Resource\_1 and Resource\_2 (Resource\_2 is brought online first) because VCS recognizes Resource\_3 is disabled. In accordance with online logic, the transaction is not propagated to the disabled resource.

In the figure below, Resource\_2 is disabled. When the service group is brought online, resources Resources 1, 3, and 4 are also brought online (Resource\_4 is brought online first). Note Resource\_3, the child of the disabled resource, is brought online because Resource\_1 is enabled and is dependent on it.



## How Disabled Resources Affect Group States

When a service group is brought online containing non-persistent, disabled resources whose AutoStart attributes are set to 1, the group state is PARTIAL, even though enabled resources with Autostart=1 are online. This is because the disabled resource is considered for the group state.

To have the group in the ONLINE state when enabled resources with AutoStart set to 1 are in ONLINE state, set the AutoStart attribute to 0 for the disabled, non-persistent resources.



## VCS Utilities

VCS provides several utilities that address common issues, however, *you must use them with extreme caution*. For best results, contact VERITAS Technical Support prior to using the utilities described below.

### The getcomms.pl Utility

The getcomms.pl utility collects and writes (“dumps”) information related to the private network. The information can be used by Technical Support to debug network-related issues.

To access getcomms.pl, type:

```
# /opt/VRTSgab/getcomms.pl -option
```

### Log Location

The getcomms.pl utility writes the output to the directory `/tmp/commslog.timestamp`. The variable *timestamp* represents the time the log was taken.

### Options for getcomms.pl

Use the following options with the `getcomms.pl` command to limit the diagnostic information to specific components.

---

Options	Action
<code>-local</code>	Retrieves and dumps information about the local system.
<code>-remote</code>	Retrieves and dumps information about all live systems in the cluster.
<code>-stuck</code>	Prints the message queue.
<code>-d logdir</code>	Dumps information at the directory specified by the variable <i>logdir</i> .

---

## The hagetcf Utility

The hagetcf utility retrieves and writes detailed diagnostic information about the VCS configuration. The information can be used by Technical Support to debug configuration-related issues.

To access hagetcf, type:

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

Running hagetcf displays output similar to the example below:

```
Enter path where configuration can be saved (default is /tmp ):
Collecting lslpp infos
Collecting nfso infos
Collecting network attributes infos
Collecting VCS information
Collecting system configuration

Collecting selected /etc files
Collecting files from /var/VRTSvcs/log
Collecting files from /var/VRTSvcs/lock
Collecting /var/adm/ras
Collecting core files in /opt/VRTSvcs
Compressing /tmp/vcsconf.thoribm29b.tar to
/tmp/vcsconf.thoribm29b.tar.gz

Done. Please e-mail /tmp/vcsconf.thoribm29b.tar.gz to your support
provider.Log Location
```

The hagetcf utility writes the output to the directory `/tmp/vcsconf.sysname.tar.gz`.



## Troubleshooting VCS Startup

This section includes error messages associated with starting VCS (shown in bold text), and provides descriptions of each error and the recommended action.

---

**“VCS:10622 local configuration missing”**

**“VCS:10623 local configuration invalid”**

**“VCS:10624 local configuration stale”**

The local configuration is invalid.

**Recommended Action:** Start the VCS engine, HAD, on another system that has a valid configuration file. The system with the configuration error “pulls” the valid configuration from the other system.

Another method is to correct the configuration file on the local system and force VCS to reread the configuration file. If the file appears valid, verify that is not an earlier version. It is possible that VCS marked the configuration stale by creating a .stale file because the last VCS shutdown was not graceful. The .stale file is created in the directory /etc/VRTSvcs/conf/config.

Type the following commands to verify the configuration and force VCS to reread the configuration file:

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

---

**“VCS:11032 registration failed. Exiting”**

GAB was not registered or has become unregistered.

**Recommended Action:** GAB is registered by the `gabconfig` command in the file /etc/gabtab. Verify that the file exists and that it contains the command `gabconfig -c`.

GAB can become unregistered if LLT is set up incorrectly. Verify that the file is correct in /etc/llttab. If the LLT configuration is incorrect, make the appropriate changes and reboot.

---

**“Waiting for cluster membership.”**

This indicates that GAB may not be seeded. If this is the case, the command `gabconfig -a` does not show any members, and the following messages may appear on the console or in the event log. See [“VCS Seeding”](#) on page 308 for details.

```
GAB: Port a registration waiting for seed port membership
GAB: Port h registration waiting for seed port membership
```



## Troubleshooting Service Groups

This section cites the most common problems associated with bringing service groups online and taking them offline. Bold text provides a description of the problem. Recommended action is also included, where applicable.

---

### System is not in RUNNING state.

**Recommended Action:** Type `hasys -display system` to verify the system is running. See [Appendix A](#) for more information on system states.

---

### Service group not configured to run on the system.

The `SystemList` attribute of the group may not contain the name of the system.

**Recommended Action:** Use the output of the command `hagrp -display service_group` to verify the system name.

---

### Service group not configured to autostart.

If the service group is not starting automatically on the system, the group may not be configured to `AutoStart`, or may not be configured to `AutoStart` on that particular system.

**Recommended Action:** Use the output of the command `hagrp -display service_group` to verify the values of the `AutoStart` and `AutoStartList` attributes.

---

### Service group is frozen.

**Recommended Action:** Use the output of the command `hagrp -display service_group` to verify the value of the `Frozen` and `TFrozen` attributes. Use the command `hagrp -unfreeze` to thaw the group. Note that VCS will not take a frozen service group offline.



### Service group autodisabled.

When VCS does not know the status of a service group on a particular system, it autodisables the service group on that system. Autodisabling occurs under the following conditions:

- When the VCS engine, HAD, is not running on the system.
- When all resources within the service group are not probed on the system.
- When a particular system is visible through disk heartbeat only.

Under these conditions, all service groups that include the system in their SystemList attribute are autodisabled. *This does not apply to systems that are powered off.*

**Recommended Action:** Use the output of the command `hagrp -display service_group` to verify the value of the AutoDisabled attribute.

**Caution:** To bring a group online manually after VCS has autodisabled the group, make sure that the group is not fully or partially active on any system that has the AutoDisabled attribute set to 1 by VCS. Specifically, verify that all resources that may be corrupted by being active on multiple systems are brought down on the designated systems. Then, clear the AutoDisabled attribute for each system:

```
# hagrp -autoenable service_group -sys system
```

---

### Failover service group is online on another system.

The group is a failover group and is online or partially online on another system.

**Recommended Action:** Use the output of the command `hagrp -display service_group` to verify the value of the State attribute. Use the command `hagrp -offline` to offline the group on another system.

---

### Service group is waiting for the resource to be brought online/taken offline.

**Recommended Action:** Review the IState attribute of all resources in the service group to locate which resource is waiting to go online (or which is waiting to be taken offline). Use the `hastatus` command to help identify the resource. See the engine and agent logs in `/var/VRTSvcs/log` for information on why the resource is unable to be brought online or be taken offline.

To clear this state, make sure all resources waiting to go online/offline do not bring themselves online/offline. Use the command `hagrp -flush` to clear the internal state of VCS. You can now bring the service group online or take it offline on another system.

**A critical resource faulted.**

Output of the command `hagrp -display service_group` indicates that the service group has faulted.

**Recommended Action:** Use the command `hares -clear` to clear the fault.

---

**Service group is waiting for a dependency to be met.**

**Recommended Action:** To see which dependencies have not been met, type `hagrp -dep service_group` to view service group dependencies, or `hares -dep resource` to view resource dependencies.

---

**Service group not fully probed.**

This occurs if the agent processes have not monitored each resource in the service group. When the VCS engine, HAD, starts, it immediately “probes” to find the initial state of all of resources. (It cannot probe if the agent is not returning a value.) A service group must be probed on all systems included in the `SystemList` attribute before VCS attempts to bring the group online as part of `AutoStart`. This ensures that even if the service group was online prior to VCS being brought up, VCS will not inadvertently bring the service group online on another system.

**Recommended Action:** Use the output of `hagrp -display service_group` to see the value of the `ProbesPending` attribute for the system’s service group. (It should be zero.) To determine which resources are not probed, verify the local `Probed` attribute for each resource on the specified system. Zero means waiting for probe result, 1 means probed, and 2 means VCS not booted. See the engine and agent logs for information.



## Troubleshooting Resources

This section cites the most common problems associated with bringing resources online and taking them offline. Bold text provides a description of the problem. Recommended action is also included, where applicable.

---

### **Service group brought online due to failover.**

VCS attempts to bring resources online that were already online on the failed system, or were in the process of going online. Each parent resource must wait for its child resources to be brought online before starting.

**Recommended Action:** Verify that the child resources are online.

---

### **Waiting for service group states.**

The state of the service group prevents VCS from bringing the resource online.

**Recommended Action:** See [Appendix A](#) for more information on states.

---

### **Waiting for child resources.**

One or more child resources of parent resource are offline.

**Recommended Action:** Bring the child resources online first.

---

### **Waiting for parent resources.**

One or more parent resources are online.

**Recommended Action:** Take the parent resources offline first.

---

### **Waiting for resource to respond.**

The resource is waiting to come online or go offline, as indicated. VCS directed the agent to run an online entry point for the resource.

**Recommended Action:** Verify the resource's IState attribute. See the engine and agent logs in `/var/VRTSvcs/engine_A.log` and `/var/VRTSvcs/agent_A.log` for information on why the resource cannot be brought online.

**Agent not running.**

The resource's agent process is not running.

**Recommended Action:** Use `hastatus -summary` to see if the agent is listed as faulted. Restart the agent:

```
# haagent -start resource_type -sys system
```

---

**Invalid agent argument list.**

The scripts are receiving incorrect arguments.

**Recommended Action:** Verify that the arguments to the scripts are correct. Use the output of `hares -display resource` to see the value of the `ArgListValues` attribute. If the `ArgList` attribute was dynamically changed, stop the agent and restart it.

To stop the agent, type:

```
# haagent -stop resource_type -sys system
```

To restart the agent, type:

```
# haagent -start resource_type -sys system
```

## Troubleshooting Notification

Occasionally you may encounter problems when using VCS notification. This section cites the most common problems and the recommended actions. Bold text provides a description of the problem.

---

**Notifier is configured but traps are not seen on SNMP console.**

**Recommended Action:** Verify the version of SNMP traps supported by the console: VCS notifier sends SNMP v2.0 traps. If you are using HP OpenView Network Node Manager as the SNMP, verify events for VCS are configured using `xnmevents`. You may also try restarting the OpenView daemon (`ovw`) if, after merging VCS events in `vcs_trapd`, the events are not listed in the OpenView Network Node Manager Event configuration.

By default, notifier assumes the community string is public. If your SNMP console was configured with a different community, reconfigure it according to the notifier configuration. See the *VERITAS Bundled Agent Reference Guide* for more information on `NotifierMngr`.



## Troubleshooting Cluster Manager (Web Console)

Occasionally you may encounter problems when using Cluster Manager (Web Console). This section cites the most common problems and the recommended actions. Bold text provides a description of the problem, and in some cases actual error messages.

---

### **Unable to log on.**

**Recommended Action:** Verify the user name exists and that the password is correct for the user name. Then verify your browser is Java, Javascript, and cookies enabled.

---

### **Unable to view Cluster Manager on a browser using the Virtual IP/port number in URL (http://[virtual\_ip:port\_number]/vcs).**

**Recommended Action:** Verify the Web server is started on a host and on the port number specified in the URL. Then verify that the ClusterService service group, which has the IP, VCSweb, and VRTSWebApp resources configured on it, is not offline or faulted on any node. If it is, use the command line to bring the group back online on at least one node.

---

### **Unable to view Cluster Manager on a browser using the HostName/port\_number in URL (http://[host\_name:port\_number]/vcs).**

**Recommended Action:** Verify that the host is running and that the Web server is started on the host and the port specified in the URL. If the host is down, access Cluster Manager (Web Console) using the URL `http://virtual_IP:port_number/vcs`. The cause of the failover should be apparent on Cluster Manager. Use Cluster Manager to administer nodes that are up and running in cluster.



**After reconfiguring virtual IP address, cannot access the Web Console using the new IP address.**

**Recommended Action:** If ClusterService service group is online, changes in resource attributes do not take effect until you take the service group offline and bring it online. Therefore, you cannot access the Web Console using the new IP address, but you can from the previous address. To reconfigure the virtual IP address:

1. Take offline the VCSweb and webip resources.
2. Change the address attribute of the webip resource.
3. Bring online the VCSweb and webip resources.

**Flashing colors appear on Netscape while switching between Cluster Manager and other open windows.**

**Recommended Action:** If there are flashes of color while viewing Cluster Manager on Netscape Navigator 4.7 or later, it is mostly likely a color-mapping issue. Set the display to 256 colors or a higher on the host machine where the GUI is being viewed to ensure best color and clarity.

**Received the following error after issuing the webgui start command:**

```
FATAL:java.net.BindException: Address already in use
java.net.BindException: Address already in use at
java.net.PlainSocketImpl.
socketBind (Native Method) at java.net.PlainSocketImpl.
bind (PlainSocketImpl.java:397).
.....
```

**Recommended Action:** Open the Task Manager on the host where you are starting Cluster Manager (Web Console) to see which processes are running. If the output indicates it is an instance of the Web server, it is most likely running on the same port where the new instance is being created. Reissue the command `webgui start -host host_name -port port_number` with a host and port that are free. You can also view the Cluster Manager instance on the port and host on which it is running.



**Received the following error after issuing the webgui stop command:**

```
java.net.ConnectException: Connection refused
at java.net.PlainSocketImpl.
socketConnect(Native Method) at java.net.PlainSocketImpl.
doConnect(PlainSocketImpl.java:312)
.....
```

**Recommended Action:** Typically, this means the Tomcat Web server was halted on that particular host and port. There is no need to reissue the stop command.

---

**“The object type specified is invalid. It should be one of cluster, group, type, resource, or system.”**

**Recommended Action:** This error (#W10002) occurs if the page URL points to a VCS object that does not exist or was deleted. If you typed the URL, verify the URL is correct. Names of VCS objects are case-sensitive: the object name in the URL must be entered in the correct case. If you clicked a link and got this error, refresh the page and retry. If you are still unsuccessful, contact VERITAS Technical Support.

---

**“The specified group does not exist or has been deleted.”**

**Recommended Action:** This error (#W10003) indicates the service group whose information you tried to access does not exist, or was deleted. If you typed the URL, verify the URL is correct. If you clicked a link to get information about the service group, verify the service group exists. Refresh the display to get current information.

---

**“The specified system does not exist or has been deleted.”**

**Recommended Action:** This error (#W10004) indicates the system whose information you tried to access does not exist, or was deleted. If you typed the URL, verify the URL is correct. If you clicked a link to get information about the system, verify the system exists. Refresh the display to get current information.

---

**“The specified resource type does not exist or has been deleted.”**

**Recommended Action:** This error (#W10005) indicates the resource type whose information you tried to access does not exist, or was deleted. If you typed the URL, verify the URL is correct. If you clicked a link to get information about the resource type, verify the resource type exists. Refresh the display to get current information.

**“The specified resource does not exist or has been deleted.”**

**Recommended Action:** This error (#W10007) indicates the resource whose information you tried to access does not exist, or was deleted. If you typed the URL, verify the URL is correct. If you clicked a link to get information about the resource type, verify the resource exists. Refresh the display to get current information.

---

**“Retrieving data from the VCS engine. Please try after some time.”**

**Recommended Action:** This error (#R10001) indicates a “snapshot” of the VCS engine, HAD, is being taken. Wait a few moments then retry the operation.

---

**“Could not log on to the VCS engine.”**

**Recommended Action:** This error (#R10002) indicates Cluster Manger (Web Console) could not connect to the VCS engine. Wait a few moments then retry the operation.

---

**“Cannot monitor VCS QuickStart.”**

**Recommended Action:** This error ( R10005) indicates you tried to connect to a cluster configured by VCS QuickStart. Cluster Manager (Web Console) cannot connect to VCS QuickStart. Use the VCS QuickStart Web graphical user interface instead.

---

**“The user could not be authenticated at this time. This could be because a snapshot of the VCS Server is being taken currently.”**

**Recommended Action:** This error (#H10001) indicates a snapshot of the VCS engine is being taken. Wait a few moments then retry the operation.

---

**“The user could not be authenticated at this time. This could be because a snapshot of the VCS Server is being taken currently.”**

**Recommended Action:** This error (#H10001) indicates a snapshot of the VCS engine is being taken. Wait a few moments then retry the operation.

---

**“The URL you specified can be accessed only if you are logged on.”**

**Recommended Action:** This error (#G10001) indicates you tried to access a page that requires authentication. Log on to VCS and retry the operation.







## Section V Appendixes

---

This section provides various appendixes containing useful, supplemental information.

Section V includes the following appendixes:

- ◆ **Appendix A “System States” 329**
- ◆ **Appendix B “VCS Attributes” 333**
- ◆ **Appendix C “Additional Administration Matrices” 365**

# System States



Whenever the VCS engine is running on a system, it is in one of the states described in the table below. States indicate a system's current mode of operation. When the engine is started on a new system, it identifies the other systems available in the cluster and their states of operation. If a cluster system is in the state of `RUNNING`, the new system retrieves the configuration information from that system. Changes made to the configuration while it is being retrieved are applied to the new system before it enters the `RUNNING` state.

If no other systems are up and in the state of `RUNNING` or `ADMIN_WAIT`, and the new system has a configuration that is not marked "stale," the engine transitions to the state `LOCAL_BUILD`, and builds the configuration from disk. If the configuration is marked "stale," the system transitions to the state of `STALE_ADMIN_WAIT`.

The following table provides a list of VCS system states and their descriptions. Examples of how states transition from one state to another begin on [page 331](#).

State	Definition
<code>ADMIN_WAIT</code>	The running configuration was lost. A system transitions into this state for the following reasons: <ul style="list-style-type: none"><li>- The last system in the <code>RUNNING</code> configuration leaves the cluster before another system takes a snapshot of its configuration and transitions to the <code>RUNNING</code> state.</li><li>- A system in <code>LOCAL_BUILD</code> state tries to build the configuration from disk and receives an unexpected error from <code>ha.cf</code> indicating the configuration is invalid.</li></ul>
<code>CURRENT_DISCOVER_WAIT</code>	The system has joined the cluster and its configuration file is valid. The system is waiting for information from other systems before it determines how to transition to another state.
<code>CURRENT_PEER_WAIT</code>	The system has a valid configuration file and another system is doing a build from disk ( <code>LOCAL_BUILD</code> ). When its peer finishes the build, this system transitions to the state <code>REMOTE_BUILD</code> .



<b>State</b>	<b>Definition</b>
EXITING	The system is leaving the cluster.
EXITED	The system has left the cluster.
EXITING_FORCIBLY	An <code>hastop -force</code> command has forced the system to leave the cluster.
FAULTED	The system has left the cluster unexpectedly.
INITING	The system has joined the cluster. This is the initial state for all systems.
LEAVING	The system is leaving the cluster gracefully. When the agents have been stopped, and when the current configuration is written to disk, the system transitions to EXITING.
LOCAL_BUILD	The system is building the running configuration from the disk configuration.
REMOTE_BUILD	The system is building a running configuration that it obtained from a peer in a RUNNING state.
RUNNING	The system is an active member of the cluster.
STALE_ADMIN_WAIT	The system has a stale configuration and there is no other system in the state of RUNNING from which to retrieve a configuration. If a system with a valid configuration is started, that system enters the LOCAL_BUILD state. Systems in STALE_ADMIN_WAIT transition to STALE_PEER_WAIT.
STALE_DISCOVER_WAIT	The system has joined the cluster with a stale configuration file. It is waiting for information from any of its peers before determining how to transition to another state.
STALE_PEER_WAIT	The system has a stale configuration file and another system is doing a build from disk (LOCAL_BUILD). When its peer finishes the build, this system transitions to the state REMOTE_BUILD.
UNKNOWN	The system has not joined the cluster because it does not have a system entry in the configuration.



---

## Examples of State Transitions

- ◆ If VCS is started on a system, and if that system is the only one in the cluster with a valid configuration, the system transitions to the `RUNNING` state:  
`INITING → CURRENT_DISCOVER_WAIT → LOCAL_BUILD → RUNNING`
- ◆ If VCS is started on a system with a valid configuration file, and if at least one other system is already in the `RUNNING` state, the new system transitions to the `RUNNING` state:  
`INITING → CURRENT_DISCOVER_WAIT → REMOTE_BUILD → RUNNING`
- ◆ If VCS is started on a system with a stale configuration file, and if at least one other system is already in the `RUNNING` state, the new system transitions to the `RUNNING` state:  
`INITING → STALE_DISCOVER_WAIT → REMOTE_BUILD → RUNNING`
- ◆ If VCS is started on a system with a stale configuration file, and if all other systems are in `STALE_ADMIN_WAIT` state, the system transitions to the `STALE_ADMIN_WAIT` state as shown below. A system stays in this state until another system with a valid configuration file is started, or when the command `hasys -force` is issued.  
`INITING → STALE_DISCOVER_WAIT → STALE_ADMIN_WAIT`
- ◆ If VCS is started on a system with a valid configuration file, and if other systems are in the `ADMIN_WAIT` state, the new system transitions to the `ADMIN_WAIT` state.  
`INITING → CURRENT_DISCOVER_WAIT → ADMIN_WAIT`
- ◆ If VCS is started on a system with a stale configuration file, and if other systems are in the `ADMIN_WAIT` state, the new system transitions to the `ADMIN_WAIT` state.  
`INITING → STALE_DISCOVER_WAIT → ADMIN_WAIT`
- ◆ When a system in `RUNNING` state is stopped with the `hasstop` command, it transitions to the `EXITED` state as shown below. During the `LEAVING` state, any online system resources are taken offline. When all of the system's resources are taken offline and the agents are stopped, the system transitions to the `EXITING` state, then `EXITED`.  
`RUNNING → LEAVING → EXITING → EXITED`





This chapter contains a comprehensive list of all VCS attributes. Attributes are categorized by cluster object, as indicated below:

- [Resource Attributes](#)
- [Resource Type Attributes](#)
- [Service Group Attributes](#)
- [System Attributes](#)
- [Cluster Attributes](#)

You can modify the values of attributes labeled “user-defined” from the command line or graphical user interface, or by manually modifying the `main.cf` configuration file. The default values of VCS attributes are suitable for most environments; however, you can change the attribute values to better suit your environment and enhance performance.

---

**Caution** When changing the values of attributes, be aware that VCS attributes interact with each other. After changing the value of an attribute, observe the cluster systems to confirm that unexpected behavior does not impair performance.

---

The values of attributes labeled “system use only” are set by VCS and are read-only. They contain important information about the state of the cluster.

The values labeled “agent-defined” are set by the corresponding agent and are also read-only.

In addition to the attributes listed in this appendix, see the *VERITAS Cluster Server Agent Developer's Guide*.



## Resource Attributes

Attributes	Type, Dimension, Scope	Definition
ArgListValues (agent-defined)	string-vector local	List of arguments passed to the resource's agent on each system. This attribute is resource- and system-specific, meaning that the list of values passed to the agent depend on which system and resource they are intended.
AutoStart (user-defined)	boolean-scalar global	Indicates the resource is brought online when the service group is brought online. Default = 1
ConfidenceLevel (agent-defined)	integer-scalar local	Indicates the level of confidence in an online resource. Values range from 0–100. Note that some VCS agents may not take advantage of this attribute and may always set it to 0. Set the level to 100 if the attribute is not used.
Critical (user-defined)	boolean-scalar global	Indicates the service group is faulted when the resource, or any resource it depends on, faults. Default = 1
Enabled (user-defined)	boolean-scalar global	Indicates agents monitor the resource. If a resource is created dynamically while VCS is running, you must enable the resource before VCS monitors it. For more information on how to add or enable resources, see the chapters on administering VCS from the command line and graphical user interfaces. When Enabled is set to 0, it implies a disabled resource. VCS will not bring a disabled resource, nor its children online, even if the children are enabled. See <a href="#">“Disabling Resources”</a> on page 311 for details. If you specify the resource in main.cf prior to starting VCS, the default value for this attribute is 1.



Attributes	Type, Dimension, Scope	Definition
Flags (system use only)	integer, scalar local	<p>Provides additional information for the state of a resource. Primarily this attribute raises flags pertaining to the resource.</p> <p>Values:</p> <p>NORMAL indicates standard working order.</p> <p>RESTARTING indicates the resource faulted and that the agent is attempting to restart the resource on the same system.</p> <p>STATE UNKNOWN indicates the latest monitor call by the agent could not determine if the resource was online or offline.</p> <p>MONITOR TIMEDOUT indicates the latest monitor call by the agent was terminated because it exceeded the maximum time specified by the static attribute MonitorTimeout.</p> <p>UNABLE TO OFFLINE indicates the agent attempted to offline the resource but the resource did not go offline. This flag is also set when a resource faults and the <code>clean</code> entry point completes successfully, but the subsequent monitor hangs or is unable to determine resource status.</p>
Group (system use only)	string-scalar global	String name of the service group to which the resource belongs.
LastOnline (system use only)	string-scalar global	Indicates the system name on which the resource was last online. This attribute is set automatically by VCS.
MonitorOnly (user-defined)	boolean-scalar global	<p>Indicates if the resource can be brought online or taken offline. If set to 0, resource can be brought online or taken offline. If set to 1, resource can be monitored only.</p> <p>Default = 0</p>



Attributes	Type, Dimension, Scope	Definition
IState (system use only)	integer-scalar local	<p>Indicates internal state of a resource. In addition to the State attribute, this attribute shows to which state the resource is transitioning.</p> <p>Values:</p> <p>NOT WAITING Resource is not in transition.</p> <p>WAITING TO GO ONLINE Agent notified to bring the resource online but procedure not yet complete.</p> <p>WAITING FOR CHILDREN ONLINE Resource to be brought online, but resource depends on at least one offline resource. Resource transitions automatically to WAITING TO GO ONLINE when all children are online.</p> <p>WAITING TO GO OFFLINE Agent notified to take the resource offline but procedure not yet complete.</p> <p>WAITING TO GO OFFLINE (propagate) Same as above, but when completed the resource's children will also be offline.</p> <p>WAITING TO GO ONLINE (reverse) Resource waiting to be brought online, but when it is online it automatically attempts to go offline. Typically this is the result of issuing an offline command while resource was waiting to go online.</p> <p>WAITING TO GO OFFLINE (reverse/propagate) Same as above, but resource propagates offlining.</p>
Name (system use only)	string-scalar global	Contains actual name of resource.
Path (user-defined)	integer-scalar local	<p>Set to 1 to identify a resource as a member of a path in the dependency tree to be taken offline on a specific system after a resource faults.</p> <p>If set to 1, this attribute applies to all resources in the upward path of the dependency tree, starting from the faulted resource.</p>



Attributes	Type, Dimension, Scope	Definition
Probed (system use only)	boolean-scalar local	Indicates whether the resource has been detected by the agent.
ResourceOwner (user-defined)	string-scalar global	<p>Used for VCS email notification and logging. VCS sends email notification to the person designated in this attribute when an event occurs related to the resource.(See “<a href="#">Notification</a>” on page 259.) VCS also logs the owner name when an event occurs.</p> <p>For example:  TAG_E 2001/12/03 11:23:48  VCS:10304:Resource file1  (Owner=Daniel, Group testgroup)  is offline on machine1</p> <p>ResourceOwner is specified as an email ID without the domain name (Daniel, not Daniel@domain.com)</p> <p>If ResourceOwner is not specified in main.cf, the default value is “unknown.”</p> <p>For example:  TAG_E 2001/12/03 11:23:48  VCS:10304:Resource file1  (Owner=unknown Group testgroup)  is offline on machine1</p>
Signaled (system use only)	integer-association local	Indicates whether a resource has been traversed. Used when bringing a service group online or taking it offline.
Start (system use only)	integer-scalar local	Indicates whether a resource was started (the process of bringing it online was initiated) on a system.



<b>Attributes</b>	<b>Type, Dimension, Scope</b>	<b>Definition</b>
State (system use only)	integer-scalar local	Resource state displays the state of the resource and the flags associated with the resource. (Flags are also captured by the Flags attribute.) This attribute and Flags present a comprehensive view of the resource's current state. Values: ONLINE OFFLINE FAULTED ONLINE   STATE UNKNOWN ONLINE   MONITOR TIMEDOUT ONLINE   UNABLE TO OFFLINE OFFLINE   STATE UNKNOWN FAULTED   RESTARTING A FAULTED resource is physically offline, though unintentionally.
TriggerEvent (system use only)	boolean-scalar global	A flag that turns Events on or off. (Currently inactive.)



## Resource Type Attributes

For more information on any attribute listed below, see the chapter on setting agent parameters in the *VERITAS Cluster Server Agent Developer's Guide*.

Attributes	Type, Dimension, Scope	Definition
AgentClass (user-defined)	string-scalar global	Indicates the scheduling class for the VCS agent process. Default = "TS"
AgentFailedOn (system use only)	string-keylist global	A list of systems on which the agent for the resource type has failed.
AgentPriority (user-defined)	string-scalar global	Indicates the priority in which the agent process runs. Default = ""
AgentReplyTimeout (user-defined)	integer-scalar global	The number of seconds the engine waits to receive a heartbeat from the agent before restarting the agent. Default = 130 seconds
AgentStartTimeout (user-defined)	integer-scalar global	The number of seconds after starting the agent that the engine waits for the initial agent "handshake" before restarting the agent. Default = 60 seconds
ArgList (user-defined)	string-vector global	An ordered list of attributes whose values are passed to the <code>open</code> , <code>close</code> , <code>online</code> , <code>offline</code> , <code>monitor</code> , and <code>clean</code> entry points.
AttrChangedTimeout (user-defined)	integer-scalar global	Maximum time (in seconds) within which the <code>attr_changed</code> entry point must complete or be terminated. Default = 60 seconds
CleanTimeout (user-defined)	integer-scalar global	Maximum time (in seconds) within which the <code>clean</code> entry point must complete or else be terminated. Default = 60 seconds



Attributes	Type, Dimension, Scope	Definition
CloseTimeout (user-defined)	integer-scalar global	Maximum time (in seconds) within which the <code>close</code> entry point must complete or else be terminated. Default = 60 seconds
ConfInterval (user-defined)	integer-scalar global	When a resource has remained online for the specified time (in seconds), previous faults and restart attempts are ignored by the agent. (See <code>ToleranceLimit</code> and <code>RestartLimit</code> attributes for details.) Default = 600 seconds
FaultOnMonitorTimeouts (user-defined)	integer-scalar global	When a monitor times out as many times as the value specified, the corresponding resource is brought down by calling the <code>clean</code> entry point. The resource is then marked <code>FAULTED</code> , or it is restarted, depending on the value set in the <code>Restart Limit</code> attribute. When <code>FaultOnMonitorTimeouts</code> is set to 0, monitor failures are not considered indicative of a resource fault. A low value may lead to spurious resource faults, especially on heavily loaded systems. Default = 4
LogFileSize (user-defined)	integer-scalar global	Specifies the size (in bytes) of the agent log file. Minimum value is 65536 bytes. Maximum value is 134217728 bytes (128MB). Default = 33554432 (32MB)
LogLevel (user-defined)	string-scalar global	Specifies the type of messages to be logged to the system's local log file. Values include: all (not recommended) debug info error none Default = error



Attributes	Type, Dimension, Scope	Definition
LogTags (user-defined)	string-scalar global	Identifies tags specific to the agent log file. Note that tags can range from A through Z; tags A through H cannot be disabled. Default = ABCDEFGH
MonitorInterval (user-defined)	integer-scalar global	Duration (in seconds) between two consecutive monitor calls for an ONLINE or transitioning resource. Default = 60 seconds A lower value may impact performance if many resources of the same type exist. A higher value may delay detection of a faulted resource.
MonitorTimeout (user-defined)	integer-scalar global	Maximum time (in seconds) within which the monitor entry point must complete or else be terminated. Default = 60 seconds
NameRule (user-defined)	string-scalar global	Generates the unique name of the resource if the resource name is not specified.
NumThreads (user-defined)	integer-scalar global	Number of threads used within the agent process for managing resources. This number does not include threads used for other internal purposes. Default = 10 Increasing to a significantly large value can degrade system performance. Decreasing to 1 prevents multiple threads. The agent framework limits the maximum value of this attribute to 20.
OfflineMonitorInterval (user-defined)	integer-scalar global	Duration (in seconds) between two consecutive monitor calls for an OFFLINE resource. If set to 0, OFFLINE resources are not monitored. Default = 300 seconds
OfflineTimeout (user-defined)	integer-scalar global	Maximum time (in seconds) within which the offline entry point must complete or else be terminated. Default = 300 seconds



Attributes	Type, Dimension, Scope	Definition
OnlineRetryLimit (user-defined)	integer-scalar global	Number of times to retry <code>online</code> , if the attempt to online a resource is unsuccessful. This parameter is meaningful only if <code>clean</code> is implemented. Default = 0
OnlineTimeout (user-defined)	integer-scalar global	Maximum time (in seconds) within which the <code>online</code> entry point must complete or else be terminated. Default = 300 seconds Increase only if resource is likely to take a longer time to come online.
OnlineWaitLimit (user-defined)	integer-scalar global	Number of monitor intervals to wait after completing the online procedure, and before the resource becomes online. Default = 2
OpenTimeout (user-defined)	integer-scalar global	Maximum time (in seconds) within which the <code>open</code> entry point must complete or else be terminated. Default = 60 seconds
Operations (user-defined)	string-scalar global	Indicates valid operations of resources of the resource type. Values are <code>OnOnly</code> (can online only), <code>OnOff</code> (can online and offline), <code>None</code> (cannot online or offline).
RestartLimit (user-defined)	integer-scalar global	Number of times to retry bringing a resource online when it is taken offline unexpectedly and before VCS declares it <code>FAULTED</code> . Default = 0
ScriptClass (user-defined)	string-scalar	Indicates the scheduling class of the script processes (for example, <code>online</code> ) created by the agent. Default = "TS"
ScriptPriority (user-defined)	string-scalar	Indicates the priority of the script processes created by the agent. Default = ""

<b>Attributes</b>	<b>Type, Dimension, Scope</b>	<b>Definition</b>
SourceFile (user-defined)	string-scalar global	File from which the configuration was read. Always set to <code>.\types.cf</code> .
ToleranceLimit (user-defined)	integer-scalar global	Number of times the <code>monitor</code> entry point should return <code>OFFLINE</code> before declaring the resource <code>FAULTED</code> . Default = 0 A large value could delay detection of a genuinely faulted resource.



## Service Group Attributes

Attributes	Type, Dimension, Scope	Definition
ActiveCount (system use only)	integer-scalar local	Number of resources in a service group that are active (online or waiting to go online). When the number drops to zero, the service group is considered offline.
Administrators (user-defined)	keylist-string global	List of VCS users with privileges to administer the group. <b>Note</b> A Group Administrator can perform all operations related to a specific service group, but cannot perform generic cluster operations. See “User Privileges” on page 73 for details. Default = ""
AutoDisabled (system use only)	boolean-scalar local (parallel groups) global (failover groups)	Indicates that VCS does not know the status of a service group (or specified system for parallel service groups). This is due to: <ul style="list-style-type: none"> <li>- Group not probed (on specified system for parallel groups) in the SystemList attribute.</li> <li>- VCS engine is not running on a node designated in the SystemList attribute, but the node is visible.</li> </ul>
AutoFailOver (user-defined)	boolean-scalar global	Indicates whether VCS initiates an automatic failover if the service group faults. Default = 1 (enabled)
AutoRestart (user-defined)	boolean-scalar global	Restarts a service group after a faulted persistent resource becomes online. See “Categories of Service Group Dependencies” on page 45 for details. <b>Note</b> This attribute applies to persistent resources only. Default = 1 (enabled)
AutoStart (user-defined)	boolean-scalar global	Designates whether a service group is automatically started when VCS is started. Default = 1 (enabled)



Attributes	Type, Dimension, Scope	Definition
AutoStartIfPartial (user-defined)	boolean-scalar global	Indicates whether to initiate bringing a service group online if the group is probed and discovered to be in a PARTIAL state when VCS is started.  Default = 1 (enabled)
AutoStartList (user-defined)	string-keylist global	List of systems on which, under specific conditions, the service group will be started with VCS (usually at system boot). For example, if a system is a member of a failover service group's AutoStartList attribute, and if it is not already running on another system in the cluster, the group is brought online when the system is started.  VCS uses the AutoStartPolicy attribute (described below) to determine the system on which to bring the service group online.  <b>Note</b> For the service group to start, AutoStart must be enabled and Frozen must be 0. Also, beginning with 1.3.0, you must define the SystemList attribute prior to setting this attribute.  Default = "" (none)
AutoStartPolicy (user-defined)	string-scalar global	Sets the policy VCS uses to determine on which system to bring a service group online if multiple systems are available.  This attribute has three options:  Order (default): Systems are chosen in the order in which they are defined in the AutoStartList attribute.  Load: Systems are chosen in the order of their capacity, as designated in the AvailableCapacity system attribute. System with the highest capacity is chosen first.  Priority: Systems are chosen in the order of their priority in the SystemList attribute. Systems with the lowest priority is chosen first.
CurrentCount (system use only)	integer-scalar global	Number of systems on which the service group is active.



Attributes	Type, Dimension, Scope	Definition
Enabled (user-defined)	boolean-scalar local (parallel groups) global (failover groups)	Indicates if a group can be failed over or brought online. If any of the local values are disabled, the group is disabled. Default = 1 (enabled)
Evacuate (user-defined)	boolean-scalar global	Indicates if VCS initiates an automatic failover when user issues <code>hastop -local -evacuate</code> . Default = 1
Evacuating (system use only)	integer-scalar global	Indicates the node ID from which the service group is being evacuated.
Failover (system use only)	boolean-scalar global	Indicates service group is in the process of failing over.
FailOverPolicy (user-defined)	string-scalar global	Sets the policy VCS uses to determine which system a group fails over to if multiple systems exist. Values: Priority (default): The system defined as the lowest priority in the SystemList attribute is chosen. Load: The system defined with the least value in the system's Load attribute is chosen. RoundRobin: Systems are chosen according to how many active service groups they are hosting. The system with the least number of active service groups is chosen first.
FromQ (system use only)	string-association global	Indicates the system name from which the service group is failing over. This attribute is specified when service group failover is a direct consequence of the group event, such as a resource fault within the group or a group switch.
Frozen (user-defined)	boolean-scalar global	Disables all actions, including autostart, online and offline, and failover, except for monitor actions performed by agents. (This convention is observed by all agents supplied with VCS.) Default = 0 (not frozen)

Attributes	Type, Dimension, Scope	Definition
GroupOwner (user-defined)	string-scalar global	<p>This attribute is used for VCS email notification and logging. VCS sends email notification to the person designated in this attribute when an event occurs related to the service group. (See <a href="#">“Notification”</a> on page 259.) VCS also logs the owner name when an event occurs.</p> <p>For example: TAG_E 2001/12/03 11:23:48 VCS:10304:Group file1 (Owner=Daniel, Group testgroup) is offline on machine1</p> <p>Group Owner is specified as an email ID without the domain name (Daniel, not Daniel@domain.com)</p> <p>If GroupOwner is not specified in main.cf, the default value is “unknown.”</p> <p>For example: TAG_E 2001/12/03 11:23:48 VCS:10304:Group file1 (Owner=unknown Group testgroup) is offline on machine1</p> <p>Default = ""</p>
IntentOnline (system use only)	integer-scalar global (failover groups) local (parallel groups)	<p>Indicates whether to keep service groups online or offline. It is set to 1 by VCS if an attempt has been made, successful or not, to online the service group. For failover groups, this attribute is set to 0 by VCS when the group is taken offline. For parallel groups, it is set to 0 for the system when the group is taken offline or when the group faults and can fail over to another system.</p>
LastSuccess (system use only)	integer-scalar global	<p>Indicates the time when service group was last brought online. (Currently inactive.)</p>
Load (user-defined)	integer-scalar global	<p>Integer value expressing total system load this group will put on a system.</p> <p>For example, the administrator may assign a value of 100 to a large production SQL and 15 to an Web server.</p> <p>Default = 0</p>



Attributes	Type, Dimension, Scope	Definition
ManualOps (user-defined)	boolean-scalar global	Indicates if manual operations are allowed on the service group. Default = 1 (enabled)
MigrateQ (system use only)	string-association global	Indicates the system from which the service group is migrating. This attribute is specified when group failover is an indirect consequence, such as system shutdown, another group faulted and is linked to this group, etc.
NumRetries (system use only)	integer-scalar local (parallel groups) global (failover groups)	Indicates the number of attempts made to bring a service group online. This attribute is used only if the attribute OnlineRetryLimit is set for the service group.
OnlineRetryInterval (user-defined)	integer-scalar global	Indicates the interval, in seconds, during which a service group that has successfully restarted on the same system and faults again should be failed over, even if the attribute OnlineRetryLimit is non-zero. This prevents a group from continuously faulting and restarting on the same system. Default = 0
OnlineRetryLimit (user-defined)	integer-scalar global	If non-zero, specifies the number of times the VCS engine tries to restart a faulted service group on the same system on which the group faulted, before it gives up and tries to fail over the group to another system. Default = 0
Operators (user-defined)	string-keylist global	List of VCS users with privileges to operate the group. A Group Operator can only perform online/offline, and temporary freeze/unfreeze operations pertaining to a specific group. See <a href="#">"User Privileges"</a> on page 73 for details. Default = ""
Parallel (user-defined)	boolean-scalar global	Indicates if service group is parallel (1) or failover (0). Default = 0



Attributes	Type, Dimension, Scope	Definition
PathCount (system use only)	integer-scalar local (parallel groups) global (failover groups)	Number of resources in path not yet taken offline. When this number drops to zero, the engine may take the entire service group offline if critical fault has occurred.
PreOffline (system use only)	boolean-scalar global	If set to 1, VCS calls a predefined script, <code>preoffline</code> , before taking the service group offline. Users can then decide whether to continue the offline procedure or cancel it. (Currently inactive.)
PreOfflining (system use only)	integer-scalar local	Indicates that VCS engine invoked the <code>preoffline</code> script; however, the script has not yet returned with group offline.
PreOnline (user-defined)	boolean-scalar global	Indicates that the VCS engine should not online a service group in response to a manual group online, group autostart, or group failover. The engine should instead call a user-defined script that checks for external conditions before bringing the group online. Default = 0
PreOnlining (system use only)	integer-scalar local	Indicates that VCS engine invoked the <code>preonline</code> script; however, the script has not yet returned with group online.
PreonlineTimeout (user-defined)	integer, scalar global	Defines the maximum amount of time the <code>preonline</code> script takes to run the command <code>hagrp -online -nopre</code> for the group. Default = 300 seconds



Attributes	Type, Dimension, Scope	Definition
Prerequisites (user-defined)	integer-association global	An unordered set of name=value pairs denoting specific resources required by a service group. If prerequisites are not met, the group cannot go online. The format for Prerequisites is: Prerequisites() = { Name=Value, name2=value2 }. Names used in setting Prerequisites are arbitrary and not obtained from the system. Coordinate name=value pairs listed in Prerequisites with the same name=value pairs in Limits(). See “ <a href="#">Limits and Prerequisites</a> ” on page 240 for details.
Priority (user-defined)	integer-scalar global	Enables users to designate and prioritize the service group. VCS does not interpret the value; rather, this attribute enables the user to configure the priority of a service group and the sequence of actions required in response to a particular event. Default = 0
PrintTree (user-defined)	boolean-scalar global	Indicates whether the resource dependency tree should be written to the configuration file. Default = 1
Probed (system use only)	boolean-scalar local	Indicates whether all enabled resources in the group have been detected by their respective agents.
ProbesPending (system use only)	integer-scalar local	The number of resources that remain to be detected by the agent on each system.
Responding (system use only)	integer-scalar local	Indicates VCS engine is responding to a failover event and is in the process of bringing the service group online or failing over the node.
Restart (system use only)	integer-scalar global	For internal use only.
SourceFile (system use only)	string-scalar global	File from which the configuration was read. Always set to ./main.cf.



Attributes	Type, Dimension, Scope	Definition
State (system use only)	integer-scalar local	<p>Group state on each system:</p> <p>OFFLINE All non-persistent resources are offline.</p> <p>ONLINE All resources whose AutoStart attribute is equal to 1 are online.</p> <p>FAULTED At least one critical resource in the group is faulted or is affected by a fault.</p> <p>PARTIAL At least one, but not all, resources with Operations=OnOff is online, and not all AutoStart resources are online.</p> <p>STARTING Group is attempting to go online.</p> <p>STOPPING Group is attempting to go offline.</p> <p>It is possible that a group state is a combination of the multiple states described above. For example,</p> <p>OFFLINE   FAULTED</p> <p>OFFLINE   STARTED</p> <p>PARTIAL   FAULTED</p> <p>PARTIAL   STARTING</p> <p>PARTIAL   STOPPING</p> <p>ONLINE   STOPPING</p>
SystemList (user-defined)	string-association global	<p>List of systems on which the service group is configured to run and their priorities. Lower numbers indicate a preference for the system as a failover target.</p> <p><b>Note</b> Beginning with 1.3.0, you must define this attribute prior to setting the AutoStartList attribute.</p> <p>Default = "" (none)</p>
SystemZones (user-defined)	integer-association global	<p>Indicates the virtual sublists within the SystemList attribute that grant priority in failing over. Values are string/integer pairs. The string key is the name of a system in the SystemList attribute, and the integer is the number of the zone. Systems with the same zone number are members of the same zone. If a service group faults on one system in a zone, it is granted priority to fail over to another system within the same zone, despite the policy granted by the FailOverPolicy attribute.</p>



<b>Attributes</b>	<b>Type, Dimension, Scope</b>	<b>Definition</b>
Tag	string-scalar global	Identifies special-purpose service groups created for specific VCS products.
TargetCount (system use only)	integer-scalar global	Indicates the number of target systems on which the service group should be brought online.
TFrozen (user-defined)	boolean-scalar global	Indicates if service groups can be brought online on the system. Groups cannot be brought online if the attribute value is 1. Default = 0 (not frozen)
ToQ (system use only)	string-association global	Indicates the node name to which the service is failing over. This attribute is specified when service group failover is a direct consequence of the group event, such as a resource fault within the group or a group switch.
TriggerEvent (system use only)	boolean-scalar global	For internal use only.
TriggerResStateChange (user-defined)	boolean-scalar global	Determines whether or not to invoke the resstatechange trigger if resource state changes. Default = 1 (enabled)
TypeDependencies (user-defined)	string-keylist global	Creates a dependency between resource types specified in the service group list, and all instances of the respective resource type. Default = ""
UserIntGlobal (user-defined)	integer-scalar global	Use this attribute for any purpose. It is not used by VCS. Default = 0
UserStrGlobal (user-defined)	string-scalar global	Use this attribute for any purpose. It is not used by VCS. Default = ""
UserIntLocal (user-defined)	integer-scalar local	Use this attribute for any purpose. It is not used by VCS. Default = 0

Attributes	Type, Dimension, Scope	Definition
UserStrLocal (user-defined)	string-scalar local	Use this attribute for any purpose. It is not used by VCS. Default = ""

## System Attributes

Attributes	Type, Dimension, Scope	Definition
AgentsStopped (system use only)	integer-scalar global	This attribute is set to 1 on a system when all agents running on the system are stopped.
AvailableCapacity (system use only)	integer-scalar global	Indicates system's available capacity when trigger is fired. If this value is negative, the argument contains the prefix % (percentage sign); for example, %-4.
Capacity (user-defined)	integer-scalar global	Value expressing total system load capacity. This value is relative to other systems in the cluster and does not reflect any real value associated with a particular system.  For example, the administrator may assign a value of 200 to a 16-processor machine and 100 to an 8-processor machine. Default = 100
ConfigBlockCount (system use only)	integer-scalar global	Number of 512-byte blocks in configuration when the system joined the cluster.
ConfigChecksum (system use only)	integer-scalar global	Sixteen-bit checksum of configuration identifying when the system joined the cluster.
ConfigDiskState (system use only)	integer-scalar global	State of configuration on the disk when the system joined the cluster.



Attributes	Type, Dimension, Scope	Definition
ConfigFile (user-defined)	string-scalar global	Directory containing the configuration files.
ConfigInfoCnt (system use only)	integer-scalar global	The count of outstanding CONFIG_INFO messages the local node expects from a new membership message. This attribute is non-zero for the brief period during which new membership is processed. When the value returns to 0, the state of all nodes in the cluster is determined.
ConfigModDate (system use only)	integer-scalar global	Last modification date of configuration when the system joined the cluster.
CPUUsage	integer-scalar local	<p>Indicates the CPU usage of the system according to CPU percentage utilization This attribute's value is valid if Enabled is set to 1 in the CPUUsageMonitoring attribute (described below).</p> <p>This value is updated when there is a change of five percent from the last indicated value.</p> <p><b>Note</b> Non-applicable for Solaris, HP-UX, AIX, and Linux users.</p>
CPUUsageMonitoring	string-association local	<p>Monitors the system's CPU usage using various factors.</p> <p>The default values for this attribute are: Enabled = 0, NotifyThreshold = 0, NotifyTimeLimit = 0, ActionThreshold = 0, ActionTimeLimit = 0, Action = NONE.</p> <p>The values for ActionTimeLimit and NotifyTimeLimit represent the time in seconds. The values for ActionThreshold and NotifyThreshold represent the threshold in terms of CPU percentage utilization.</p> <p><b>Note</b> Non-applicable for Solaris, HP-UX, AIX, and Linux users.</p>
CurrentLimits (system use only)	integer-association global	<p>System-maintained calculation of current value of Limits.</p> <p>CurrentLimits = Limits - (additive value of all service group Prerequisites).</p>



Attributes	Type, Dimension, Scope	Definition
DiskHbStatus (system use only)	string-association global	Indicates status of communication disks on any system.
DynamicLoad (system use only)	integer-scalar global	System-maintained value of current dynamic load. The value is set external to VCS with the <code>hasys -load</code> command.
Frozen (user-defined)	boolean-scalar global	Indicates if service groups can be brought online on the system. Groups cannot be brought online if the attribute value is 1.
GUIPAddr (user-defined)	string-scalar global	Determines the local IP address that VCS uses to accept connections. Incoming connections over other IP addresses are dropped. If GUIPAddr is not set, the default behavior is to accept external connections over all configured local IP addresses.  For additional information, see “ <a href="#">User Privileges for CLI and Cluster Shell Commands</a> ” on page 82.
Limits (user-defined)	integer-association global	An unordered set of name=value pairs denoting specific resources available on a system. Names are arbitrary and are set by the administrator for any value. Names are not obtained from the system.  The format for Limits is: Limits = { Name=Value, Name2=Value2 }. Default = ""
LinkHbStatus (system use only)	string-association global	Indicates status of private network links on any system.
LLTNodeId (system use only)	integer-scalar global	Displays the node ID defined in the file <code>/etc/llttab</code> .
LoadTimeCounter (system use only)	integer-scalar global	System-maintained internal counter of how many seconds the system load has been above <code>LoadWarningLevel</code> . This value resets to zero anytime system load drops below the value in <code>LoadWarningLevel</code> .



Attributes	Type, Dimension, Scope	Definition
LoadTimeThreshold (user-defined)	integer-scalar global	How long the system load must remain at or above LoadWarningLevel before the LoadWarning trigger is fired. If set to 0 overload calculations are disabled. Default = 600 seconds
LoadWarningLevel (user-defined)	integer-scalar global	A value expressed as a percentage of total capacity where load has reached a critical limit. If set to 0 overload calculations are disabled. For example, setting LoadWarningLevel = 80 sets the warning level to 80 percent. The value of this attribute can be set from 1 to 100. If set to 1, system load must equal 1 percent of system capacity to begin incrementing the LoadTimeCounter. If set to 100, system load must equal system capacity to increment the LoadTimeCounter. Default = 80 percent
MajorVersion (system use only)	integer-scalar global	Major version of system's join protocol.
MinorVersion (system use only)	integer-scalar global	Minor version of system's join protocol.
NodeId (system use only)	integer-scalar global	System (node) identification specified in /etc/llttab.
OnGrpCnt (system use only)	integer-scalar global	Number of groups that are online, or about to go online, on a system.



Attributes	Type, Dimension, Scope	Definition
ShutdownTimeout (user-defined)	integer-scalar global	<p>Determines whether to treat system reboot as a fault for service groups running on the system.</p> <p>On many systems, when a reboot occurs the processes are stopped first, then the system goes down. When the VCS engine is stopped, service groups that include the failed system in their SystemList attributes are autodisabled. However, if the system goes down within the number of seconds designated in ShutdownTimeout, service groups previously online on the failed system are treated as faulted and failed over.</p> <p>If you do not want to treat the system reboot as a fault, set the value for this attribute to 0.</p> <p>Default = 120 seconds</p>
SourceFile (user-defined)	string-scalar global	File from which the configuration was read. Always set to <code>./main.cf</code> .
SysInfo (system use only)	string-scalar global	Provides platform-specific information, including the name, version, and release of the operating system, the name of the system on which it is running, and the hardware type.
SysName	string-scalar global	Denotes system name.
SystemLocation (user-defined)	string-scalar global	Denotes the location of the system.



Attributes	Type, Dimension, Scope	Definition
SystemOwner (user-defined)	string-scalar global	<p>This attribute is used for VCS email notification and logging. VCS sends email notification to the person designated in this attribute when an event occurs related to the system.(See “<a href="#">Notification</a>” on page 259.) VCS also logs the owner name when an event occurs.</p> <p>For example:  TAG_E 2001/12/03 11:23:48  VCS:10304:System file1  (Owner=Daniel, Group testgroup)  is offline on machine1</p> <p>SystemOwner is specified as an email ID without the domain name (Daniel, not Daniel@domain.com)</p> <p>If SystemOwner is not specified in main.cf, the default value is “unknown.”</p> <p>For example:  TAG_E 2001/12/03 11:23:48  VCS:10304:System file1  (Owner=unknown Group testgroup)  is offline on machine1</p>
SysState (system use only)	integer-scalar global	System state. (See “ <a href="#">System States</a> ” on page 329 for more information.)
TFrozen (user-defined)	boolean-scalar global	Indicates if a group can be brought online or taken offline. Default = 0
TRSE (system use only)	integer-scalar global	Indicates in seconds the time to Regular State Exit. Time is calculated as the duration between the events of VCS losing port h membership and of VCS losing port a membership of GAB.

Attributes	Type, Dimension, Scope	Definition
UpDownState (system use only)	integer-scalar global	<p>This attribute has four values:</p> <p>DOWN: System is powered off, or GAB and LLT are not running on the system.</p> <p>UP BUT NOT IN CLUSTER MEMBERSHIP:</p> <ul style="list-style-type: none"> <li>- GAB and LLT are running but the VCS engine is not.</li> <li>- The system is recognized through disk heartbeat only.</li> </ul> <p>UP AND IN JEOPARDY: The system is up and part of cluster membership, but only one network link (LLT) remains.</p> <p>UP: The system is up and part of cluster membership, and has at least two links to the cluster.</p>
UserInt (user-defined)	integer-scalar global	Stores a system's integer value. Default = 0
UserStr (user-defined)	string-scalar global	Stores a system's string value. Default = ""



## Cluster Attributes

Attributes	Type, Dimension, Scope	Definition
Administrators (user-defined)	keylist-string global	Contains list of users with Administrator privileges. See “ <a href="#">Controlling Access to VCS</a> ” on page 73. Default = ""
AllowNativeCliUsers (user-defined)	boolean-scalar global	If user does not have root privileges, and if this attribute is set to 0, user is prompted for a password when issuing haxxx commands. If this attribute is set to 1, the user is not prompted. VCS validates OS user’s login against VCS’ list of user IDs and assigns appropriate privileges. Default = 0
ClusterLocation (user-defined)	string-scalar global	Specifies the location of the cluster. Default = ""
ClusterName (user-defined)	string-scalar global	Arbitrary string containing the name of cluster. Default = ""



Attributes	Type, Dimension, Scope	Definition
ClusterOwner (user-defined)	string-scalar global	<p>This attribute is used for VCS notification; specifically, VCS sends notifications to persons designated in this attribute when something goes wrong with the cluster. See “<a href="#">Notification</a>” on page 259 for more information.</p> <p>Cluster state change messages resemble:  TAG_E 2001/12/03 11:23:48  VCS:10304:Cluster file1 (Owner=Daniel Group testgroup) is offline on thor80.</p> <p>ClusterOwner is specified as an email ID without the domain name (Daniel, not Daniel@domain.com)</p> <p>If ClusterOwner is not specified in main.cf, the default value is “unknown.” For example:  TAG_E 2001/12/03 11:23:48  VCS:10304:Cluster file1  (Owner=unknown Group testgroup) is offline on thor80</p>
CompareRSM (user-defined)	integer-scalar global	<p>Indicates if VCS engine is to verify that replicated state machine is consistent. This can be set by running the haddebug command.</p> <p>Default = 0</p>
CounterInterval (user-defined)	integer-scalar global	<p>Intervals counted by the attribute GlobalCounter indicating approximately how often a broadcast occurs that will cause the GlobalCounter attribute to increase.</p> <p>The default value of the GlobalCounter increment can be modified by changing CounterInterval. If you increase this attribute to exceed five seconds, consider increasing the default value of the ShutdownTimeout attribute.</p> <p>Default = 5</p>
DumpingMembership (system use only)	integer-scalar global	<p>Indicates that the engine is writing to disk.</p>



Attributes	Type, Dimension, Scope	Definition
EngineClass (user-defined)	string-scalar global	Indicates the scheduling class for the VCS engine (HAD). Default = "RT" <b>Note</b> Non-applicable for Windows.
EnginePriority (user-defined)	string-scalar global	Indicates the priority in which HAD runs.
GlobalCounter (system use only)	integer-scalar global	This counter increases incrementally by one for each counter interval. It increases when the broadcast is received.  VCS uses the GlobalCounter attribute to measure the time it takes to shut down a system. By default, the GlobalCounter attribute is updated every five seconds. This default value, combined with the 60-second default value of ShutdownTimeout, means if system goes down within twelve increments of GlobalCounter, it is treated as a fault. The default value of GlobalCounter increment can be modified by changing the CounterInterval attribute.
GroupLimit (user-defined)	integer-scalar global	Maximum number of service groups. Default = 200
HacliUserLevel (user-defined)	string-scalar global	This attribute has three, case-sensitive values: NONE–hacli is disabled for all users regardless of category. COMMANDROOT–hacli is enabled for root only. CLUSTERADMIN–hacli is enabled for Cluster Administrator and root. Default = NONE <b>Note</b> The command <code>haclus -modify HacliUserLevel</code> can be executed by root only.



Attributes	Type, Dimension, Scope	Definition
LockMemory (user-defined)  <b>Note</b> The attribute is system-defined for AIX.	string-scalar global	Controls the locking of VCS engine pages in memory. This attribute has three values. Values are case-sensitive: ALL: Locks all current and future pages. CURRENT: Locks current pages. NONE: Does not lock any pages. Default = ALL <b>Note</b> For AIX, this attribute includes only one value, ALL, which is the default. This value locks text and data into memory (process lock).
LogSize (user-defined)	integer-scalar global	Size of engine log file. Minimum value = 64KB Maximum value = 128MB Default value = 32MB
MajorVersion (system use only)	integer-scalar global	Major version of system's join protocol.
MinorVersion (system use only)	integer-scalar global	Minor version of system's join protocol.
Notifier (system use only)	string-association global	Indicates the status of the notifier in the cluster; specifically: State, which describes current state of notifier, such as whether or not it is connected to VCS. Host, which denotes the host on which notifier is currently running or was last running. Default = None Severity, which denotes the severity level of messages queued by VCS for notifier. Messages as severe or more severe as assigned value are queued by VCS. Values include Information, Warning, Error, and SevereError. Default = Warning Queue, which shows the current size of message queue for messages queued by VCS for notifier.



Attributes	Type, Dimension, Scope	Definition
Operators (user-defined)	keylist-string global	Contains list of users with Cluster Operator privileges. See “ <a href="#">Controlling Access to VCS</a> ” on page 73. Default = ""
ProcessClass (user-defined)	string-scalar global	Indicates the scheduling class for HAD processes (for example, triggers). Default = "TS" <b>Note</b> Non-applicable for Windows.
ProcessPriority (user-defined)	string-scalar global	Indicates the priority of HAD processes (for example, triggers). Default = ""
PrintMsg (user-defined)	boolean-scalar global	If set to 1, enables logging TagM messages in engine log. Default = 0
ReadOnly (user-defined)	integer-scalar global	Indicates that cluster is in read-only mode. Default = 1
ResourceLimit (user-defined)	integer-scalar global	Maximum number of resources. Default = 5000
SourceFile (user-defined)	string-scalar global	File from which the configuration was read. Always set to ./main.cf.
TypeLimit (user-defined)	integer-scalar global	Maximum number of resource types. Default = 100
UserNames (user-defined)	string-association global	List of VCS user names. Default = "" <b>Note</b> Default user name is “admin.”
VCSMode (system use only)	string-scalar global	Denotes the mode for which VCS is licensed, including VCS, VCS_QUICKSTART, and VCS_OPS.

# Additional Administration Matrices

C

Please note that some VCS commands can be executed from the command line only by users with root/Administrator privileges. These include:

- ◆ hacf
- ◆ had
- ◆ hagetcf
- ◆ hanotify
- ◆ hastop

## haagent

haagent Options	Guest	Group Operator	Group Admin.	Cluster Operator	Cluster Admin.
-start	-	-	-	✓	✓
-stop	-	-	-	✓	✓
-display	✓	✓	✓	✓	✓
-list	✓	✓	✓	✓	✓
-value	✓	✓	✓	✓	✓
-help	✓	✓	✓	✓	✓



## haattr

haattr Options	Guest	Group Operator	Group Admin.	Cluster Operator	Cluster Admin.
-add	-	-	-	-	✓
-add -static	-	-	-	-	✓
-default	-	-	-	-	✓
-delete -static	-	-	-	-	✓
-display	✓	✓	✓	✓	✓
-help	✓	✓	✓	✓	✓

## hacli

Do not use hacli to invoke a command on a remote system that requires user input. The process can hang and consume resources.

hacli Options	Guest	Group Operator	Group Admin.	Cluster Operator	Cluster Admin.
-cmd	-	-	-	-	-
-help	✓	✓	✓	✓	✓

## haconf

haconf Options	Guest	Group Operator	Group Admin.	Cluster Operator	Cluster Admin.
-makerw	-	-	✓	-	✓
-dump	-	-	✓	-	✓
-dump -makero	-	-	✓	-	✓
-help	✓	✓	✓	✓	✓

## hadebug

hadebug Options	Guest	Group Operator	Group Admin.	Cluster Operator	Cluster Admin.
-handle	-	-	-	-	-
-hash	-	-	-	-	-
-memory	-	-	-	-	-
-ping	✓	✓	✓	✓	✓
-startmatch	-	-	-	-	-
-stopmatch	-	-	-	-	-
-time	-	-	-	-	-
-help	✓	✓	✓	✓	✓



## halog

halog Options	Guest	Group Operator	Group Admin.	Cluster Operator	Cluster Admin.
-addtags	-	-	-	-	✓
-deltags	-	-	-	-	✓
-add	-	-	-	-	✓
-cache	✓	✓	✓	✓	✓
info	✓	✓	✓	✓	✓

## hareg

hareg Options	Guest	Group Operator	Group Admin.	Cluster Operator	Cluster Admin.
-clus	✓	✓	✓	✓	✓
-sys	✓	✓	✓	✓	✓
-group	✓	✓	✓	✓	✓
-type	✓	✓	✓	✓	✓
-attr	✓	✓	✓	✓	✓
-event	✓	✓	✓	✓	✓
-resource	✓	✓	✓	✓	✓
-groupresources	✓	✓	✓	✓	✓
-typeresources	✓	✓	✓	✓	✓
-cache	✓	✓	✓	✓	✓

# hatype

hatype Options	Guest	Group Operator	Group Admin.	Cluster Operator	Cluster Admin.
-add	-	-	-	-	✓
-delete	-	-	-	-	✓
-display	✓	✓	✓	✓	✓
-resources	✓	✓	✓	✓	✓
-modify	-	-	-	-	✓
-modify -add	-	-	-	-	✓
-modify -delete	-	-	-	-	✓
-modify -delete -keys	-	-	-	-	✓
-modify -update	-	-	-	-	✓
-list	✓	✓	✓	✓	✓
-value	✓	✓	✓	✓	✓
-help	✓	✓	✓	✓	✓





# Index

---

## A

- Agent log
  - format 293
  - location 293
- Agents
  - classifications of 17
  - conditional statements 98
  - definition of 16
  - impact on performance 283
  - start/stop manually with command line 112
- All Attributes page, Web Console 208
- All Groups page, Web Console 206
- All Systems page, Web Console 206
- All Types page, Web Console 207
- Association attribute dimension 38
- Asymmetric configuration 7
- Attribute dimensions
  - association 38
  - keylist 38
  - scalar 38
  - vector 38
- Attribute types
  - boolean 37
  - integer 37
  - string 37
- Attributes
  - editing with Java Console 199
  - editing with Web Console 230
  - local and global 40
  - monitoring with Java Console 131
  - resource-specific 39
  - type-dependent 38
  - type-independent 38
  - type-specific 39

## B

- Binary message catalogs
  - definition of 294
  - location of 294
- Boolean attribute type 37

## C

- Cluster Administrator, definition of 74
- Cluster attributes
  - Administrators 360
  - AllowNativeCliUsers 360
  - ClusterLocation 360
  - ClusterName 360
  - ClusterOwner 361
  - CompareRSM 361
  - CounterInterval 361
  - DumpingMembership 361
  - EngineClass 362
  - EnginePriority 362
  - GlobalCounter 362
  - GroupLimit 362
  - HacliUserLevel 362
  - LockMemory 363
  - LogSize 363
  - MajorVersion 363
  - MinorVersion 363
  - Notifier 363
  - Operators 364
  - PrintMsg 364
  - ProcessClass 364
  - ProcessPriority 364
  - ReadOnly 364
  - ResourceLimit 364
  - SourceFile 364
  - TypeLimit 364
  - UserNames 364
  - VCSMode 364



- Cluster communications
  - GAB 19
  - HAD 20
  - network/disk channels 21
  - overview 18
- Cluster Explorer
  - accessing 127
  - accessing Cluster Shell 140
  - accessing Command Center 141
  - accessing Heartbeat View 135
  - accessing Log Desk 145
  - accessing Properties View 131
  - accessing Resource View 133
  - accessing Service Group View 132
  - accessing Status View 130
  - accessing System Manager 138
  - configuration tree 128
  - creating tear-off views 129
  - modifying group's system list 138
  - overview 126
  - using configuration wizard 172
  - using toolbar 127
  - using view panel 129
  - using zoom panel 134
- Cluster Guest, definition of 75
- Cluster Manager
  - description 117
  - impact on performance 282
  - setting initial display 118
  - starting 118
- Cluster Monitor
  - collapsing display 122
  - configuring existing panel 147
  - configuring new panel 146
  - customizing Java Console display 123
  - expanding display 122
  - logging off of cluster 149
  - logging on to cluster 148
  - pausing scroll panel 122
  - using menus 120
  - using toolbar 120
- Cluster Operator, definition of 74
- Cluster Shell
  - accessing 140
  - definition of 140
- Cluster Summary Page, Web Console 205
- Clustering
  - criteria for data storage 4
  - criteria for monitor procedure 3
  - criteria for start procedure 3
  - criteria for stop procedure 3
  - license and host name issues 5
  - restarting application to known state 4
- Clusters
  - adding with Java Console 146
  - configuring with Java Console 147
  - logging off with Cluster Monitor 149
  - logging on with Cluster Monitor 148
  - modifying attributes with command line 102
  - opening configuration with Web Console 220
  - saving configuration with Web Console 220
  - seeding 308
  - when a system is booted 285
- Command Center
  - accessing 141
  - using Configuration folder 142
  - using Operations folder 143
- Command list
  - hastart 85, 86
- Conditional statements, definition of 98
- Configuration Editor
  - accessing System Manager 138
  - modifying system list 138
  - starting 118
  - using wizards 172
- Configuration files
  - designating as stale with command line 89
  - read/write to read-only 89
  - read-only 89
  - removing stale designation with command line 89
  - using -modify option 103
- Configuration language
  - local and global attributes 40
  - type-specific attributes 39
- Configurations
  - asymmetric 7
  - N+1 10
  - N-to-1 9



- N-to-N 10
  - symmetric 8
- CPUUsage event trigger 272
- D**
  - Dependencies
    - removing between resources with command line 110
    - removing between service groups with command line 110
- E**
  - Engine log
    - format 293
    - location 293
  - Entry points, modifying for performance 284
  - Environment variables 83
  - Error messages
    - agent log 293
    - at startup 316
    - engine log 293
    - message catalogs 294
    - Web Console 322
  - Event triggers
    - CPUUsage 272
    - definition of 271
    - hatrigger 271
    - InJeopardy 272
    - LoadWarning 273
    - NoFailover 274
    - PostOffline 275
    - PostOnline 276
    - PreOnline 280
    - ResFault 277
    - ResNotOff 278
    - ResStateChange 279
- F**
  - Failover groups 15
  - Failover vs. switchover 6
- G**
  - GAB
    - definition of 19
    - impact on performance 281
    - when a system panics 289
  - GABDISK 21
  - getcomms.pl utility 314
  - Global attributes 40
  - Group Administrator, definition of 74
  - Group Operator, definition of 74
- H**
  - haagent -display command 94
  - haagent -list command 98
  - haagent -start command 112
  - haagent -stop command 112
  - haattr -add command 111
  - haattr -default command 112
  - haattr -delete command 111
  - hacf utility 25
    - creating multiple .cf files 42
    - definition of 41
    - dumping a configuration 42
    - loading a configuration 42
    - pretty-printing 41
  - haclus -display command 95
  - haclus -modify command 90
  - haclus -value command 95
  - haconf -dump -makero 89
  - haconf -makerw command 89
  - HAD
    - definition of 20
    - impact on performance 282
  - hagetcf utility 315
  - hagrp -add command 106
  - hagrp -clear command 100
  - hagrp -delete command 110
  - hagrp -dep command 92
  - hagrp -disable command 99
  - hagrp -disableresources command 100
  - hagrp -display command 93
  - hagrp -enable command 99
  - hagrp -enableresources command 99
  - hagrp -freeze command 99
  - hagrp -list command 98
  - hagrp -modify command 91, 106
  - hagrp -offline command 99
  - hagrp -online command 99
  - hagrp -resources command 92
  - hagrp -state command 92
  - hagrp -switch command 99
  - hagrp -unfreeze command 99
  - hagrp -unlink command 110
  - halic command 85
  - halic utility 85
  - hamsg -help command 96
  - hamsg -info command 96



- hamsg -list command 96
- hanotify utility 263
- hares -add command 108
- hares -clear command 101
- hares -delete command 110
- hares -dep command 93
- hares -display command 93
- hares -global command 93
- hares -link command 109
- hares -list command 98
- hares -local command 104
- hares -modify command 109
- hares -offline command 100
- hares -offprop command 100
- hares -online command 100
- hares -probe command 100
- hares -unlink command 110
- hashadow process 25
- hastatus command 95
- hastatus -group command 95
- hastatus -summary command 95
- hasys -display command 95
- hasys -force command 101
- hasys -freeze command 102
- hasys -list command 95
- hasys -modify command 101
- hasys -nodeid command 101
- hasys -unfreeze command 102
- hatype -add command 111
- hatype -delete command 111
- hatype -display command 94
- hatype -list command 94
- hatype -modify command 111
- hatype -resources command 94
- hauser -add command 90
- hauser -display command 92
- hauser -pdate command 91
- HP OpenView Network Node Manager (NNM) 268

## I

- Include clauses, definition of 30
- InJeopardy event trigger 272
- Integer attribute type 37

## J

- Java Console
  - accessing Cluster Explorer 127
  - accessing cluster status 130
  - adding clusters 146

- adding resources 174
- adding service groups 154
- adding systems 189
- adding users 150
- administering configuration 196
- autoenabling service groups 166
- bringing groups online 158
- bringing resources online 179
- changing user password 152
- changing user privileges 153
- clearing resources 183
- closing configuration 195
- Cluster Explorer 126
- Cluster Explorer configuration tree 128
- Cluster Explorer toolbar 127
- Cluster Explorer view panel 129
- configuring clusters 147
- creating tear-off views 129
- customizing display 123
- deleting resources 178
- deleting service groups 157
- deleting systems 190
- deleting users 151
- disabling resources 182
- disabling service groups 165
- displaying log messages 145
- editing attributes 199
- enabling resources 182
- enabling service groups 164
- flushing service groups 167
- freezing service groups 162
- freezing systems 191
- importing resource types 188
- linking resources 184
- linking service groups 168
- log messages 201
- monitoring attributes 131
- monitoring group dependencies 132
- monitoring resource dependencies 133
- monitoring resources 133
- monitoring service groups 132
- opening configuration 193
- probing resources 181
- running commands 141, 198
- saving configuration 194
- setting initial display 118
- switching service groups 161
- taking groups offline 160
- taking resources offline 180



- unfreezing service groups 163
  - unfreezing systems 192
  - unlinking resources 186
  - unlinking service groups 169
  - Jeopardy, definition of 296
- K**
- Keylist attribute dimension 38
  - Keywords, list of 41
- L**
- License key
    - definition of 85
    - installing 85
  - LLT
    - disk heartbeats 21
    - low-priority link 21
  - Load policy for SGWM 238
  - LoadWarning event trigger 273
  - Local attributes 40
  - Log Desk
    - accessing 145
    - customizing a display 201
  - Log Details page, Web Console 209
  - Log display
    - editing with Web Console 233
    - resetting with Web Console 234
  - Logging
    - agent log 293
    - engine log 293
    - message tags 293
- M**
- main.cf 30
    - cluster definition 30
    - group dependency clause 32
    - include clauses 30
    - resource definition 31
    - resource dependency clause 32
    - sample configuration 32
    - service group definition 31
  - Message tags, definition of 293
- N**
- N+1 configuration 10
  - Network failure, detecting 288
  - Network partitions 295
  - NoFailover event trigger 274
- Notification**
- configuring 269
  - definition of 259
  - deleting messages 261
  - error messages 260
  - error severity levels 260
  - event and trap list 264
  - event triggers 271
  - hanotify utility 263
  - message queue 261
  - notifier process 262
  - SNMP files 268
  - troubleshooting 321
- Notifier process 262**
- N-to-1 configuration 9**
- N-to-N configuration 10**
- NumThreads attribute, modifying for performance 283**
- O**
- Offline group dependency 45
  - Online group dependency 45
  - Overload warning for SGWM 239
- P**
- Parallel groups 15
  - Password
    - authenticating 82
    - changing with Java Console 152
    - character limitation 148
  - Performance
    - agents 283
    - Cluster Manager 282
    - GAB 281
    - HAD 282
    - impact of VCS 281
    - modifying entry points 284
    - modifying NumThreads attribute 283
    - when a cluster is booted 285
    - when a resource fails 287
    - when a resource is brought online 285
    - when a resource is taken offline 286
    - when a service group fails over 290
    - when a service group is brought online 286
    - when a service group is taken offline 286
    - when a system fails 288
  - PostOffline event trigger 275



PostOnline event trigger 276  
PreOnline event trigger 280  
Priority policy for SGWM 237

## R

Reserved words, list of 41  
ResFault event trigger 277  
ResNotOff event trigger 278  
Resource attributes  
    ArgListValues 334  
    AutoStart 334  
    ConfidenceLevel 334  
    Critical 334  
    Enabled 334  
    Flags 335  
    Group 335  
    Istate 336  
    LastOnline 335  
    MonitorOnly 335  
    Name 336  
    Path 336  
    Probed 337  
    ResourceOwner 337  
    Signaled 337  
    Start 337  
    State 338  
    TriggerEvent 338  
Resource dependencies  
    child 14  
    monitoring with Java Console 133  
    parent 14  
Resource Dependency page, Web Console 214  
Resource page, Web Console 213  
Resource Type attributes  
    AgentClass 339  
    AgentFailedOn 339  
    AgentPriority 339  
    AgentReplyTimeout 339  
    AgentStartTimeout 339  
    ArgList 339  
    AttrChangedTimeout 339  
    CleanTimeout 339  
    CloseTimeout 340  
    ConfInterval 340  
    FaultOnMonitorTimeouts 340  
    LogFileSize 340  
    LogLevel 340  
    LogTags 341

MonitorInterval 341  
MonitorTimeout 341  
NameRule 341  
NumThreads 341  
OfflineMonitorInterval 341  
OfflineTimeout 341  
OnlineRetryLimit 342  
OnlineTimeout 342  
OnlineWaitLimit 342  
OpenTimeout 342  
Operations 342  
RestartLimit 342  
ScriptClass 342  
ScriptPriority 342  
SourceFile 343  
ToleranceLimit 343  
Resource Type page, Web Console 212  
Resource types  
    definition of 12  
    importing with Java Console 188  
    querying with command line 94  
Resource View  
    using zoom panel 134  
Resource View, accessing in Web Console 133  
Resources  
    adding from Template view 176  
    adding with command line 108  
    adding with Java Console 174  
    bringing online with Java Console 179  
    bringing online with Web Console 226  
    clearing with Java Console 183  
    clearing with Web Console 227  
    conditional statements 98  
    definition of 12  
    deleting with command line 110  
    deleting with Java Console 178  
    detecting failure 287  
    disabling using command line 311  
    disabling with Java Console 182  
    enabling with Java Console 182  
    how disabling affects states 313  
    limitations of disabling 311  
    linking using command line 109  
    linking using Java Console 184  
    localizing attributes with command line 104  
    monitoring with Java Console 133  
    probing with Java Console 181



---

- probing with Web Console 227
- querying with command line 93
- taking offline with Java Console 180
- taking offline with Web Console 226
- troubleshooting 320
- unlinking using Java Console 186

Resource-specific attributes 39

ResStateChange event trigger 279

Round Robin policy for SGWM 238

## S

Scalar attribute dimension 38

Seeding 308

- automatically 308
- definition of 308
- manually 308

Service group attributes

- ActiveCount 344
- Administrators 344
- AutoDisabled 344
- AutoFailover 344
- AutoRestart 344
- AutoStart 344
- AutoStartIfPartial 345
- AutoStartList 345
- AutoStartPolicy 345
- CurrentCount 345
- Enabled 346
- Evacuate 346
- Evacuating 346
- Failover 346
- FailOverPolicy 346
- FromQ 346
- Frozen 346
- GroupOwner 347
- IntentOnline 347
- LastSuccess 347
- Load 347
- ManualOps 348
- MigrateQ 348
- NumRetries 348
- OnlineRetryInterval 348
- OnlineRetryLimit 348
- Operators 348
- Parallel 348
- PathCount 349
- PreOffline 349
- PreOfflining 349
- PreOnline 349

- PreonlineTimeout 349
- PreOnlining 349
- Prerequisites 350
- PrintTree 350
- Priority 350
- Probed 350
- ProbesPending 350
- Responding 350
- Restart 350
- SourceFile 350
- State 351
- SystemList 351
- SystemZones 351
- Tag 352
- TargetCount 352
- TFrozen 352
- ToQ 352
- TriggerEvent 352
- TriggerResStateChange 352
- TypeDependencies 352
- UserIntGlobal 352
- UserIntLocal 352
- UserStrGlobal 352
- UserStrLocal 353

Service group dependencies

- automatic failover 62
- automatic online 61
- autorestart 61
- configuring 44, 60
- definition of 43
- failover parent/failover child 52
- failover parent/parallel child 55
- FAQs 66
- firm 50
- global 48
- limitations 60
- linking 64
- local 47
- locations of 47
- manual offline 64
- manual online 63
- manual switch 64
- monitoring with Java Console 132
- offline group dependency 45
- online group dependency 45
- parallel parent/failover child 57
- parallel parent/parallel child 59
- remote 48
- soft 49



- summary sheet 66
- types of 49
- Service Group page, Web Console 210
- Service group workload management
  - Capacity and Load attributes 238
  - definition of 237
  - Limits and Prerequisites attributes 240
  - load policy 238
  - Load-based autostart 242
  - overload warning 239
  - priority policy 237
  - round robin policy 238
  - sample configurations 243
  - SystemZones attribute 241
- Service groups
  - adding with command line 106
  - adding with Java Console 154
  - administering with command line 99
  - autoenabling with Java Console 166
  - autoenabling with Web Console 224
  - bringing online with Java Console 158
  - bringing online with Web Console 221
  - conditional statements 98
  - definition of 13
  - deleting with command line 110
  - deleting with Java Console 157
  - disabling with Java Console 165
  - disabling with Web Console 223
  - enabling with Java Console 164
  - enabling with Web Console 223
  - flushing with Java Console 167
  - flushing with Web Console 225
  - freezing with Java Console 162
  - freezing with Web Console 224
  - linking using Java Console 168
  - modifying list using Cluster Explorer 138
  - modifying system list with Configuration Editor 138
  - modifying system lists with command line 107
  - modifying with command line 106
  - monitoring with Java Console 132
  - querying with command line 92
  - switching using Java Console 161
  - switching using Web Console 222
  - taking offline with Java Console 160
  - taking offline with Web Console 222
  - troubleshooting 317
  - types of 15
  - unfreezing with Java Console 163
  - unfreezing with Web Console 225
  - unlinking using Java Console 169
  - workload management 237
- SMTP 259
- SNMP 259
  - files for notification 268
  - HP OpenView 268
  - merging events with HP OpenView NNM 268
- Split brain 296
- String attribute type 37
- Switchover, definition of 6
- Symmetric configuration 8
- System attributes
  - AgentStopped 353
  - AvailableCapacity 353
  - Capacity 353
  - ConfigBlockCount 353
  - ConfigChecksum 353
  - ConfigDiskState 353
  - ConfigFile 354
  - ConfigInfoCnt 354
  - ConfigModDate 354
  - CPUUsage 354
  - CPUUsageMonitoring 354
  - CurrentLimits 354
  - DiskHbStatus 355
  - DynamicLoad 355
  - Frozen 355
  - GUIPAddr 355
  - Limits 355
  - LinkHbStatus 355
  - LLTNodeId 355
  - LoadTimeCounter 355
  - LoadTimeThreshold 356
  - LoadWarningLevel 356
  - MajorVersion 356
  - MinorVersion 356
  - NodeId 356
  - OnGrpCnt 356
  - ShutdownTimeout 357
  - SourceFile 357
  - SysInfo 357
  - SysName 357
  - SysState 358
  - SystemLocation 357
  - SystemOwner 358



- TFrozen 358
- TRSE 358
- UpDownState 359
- UserInt 359
- UserStr 359
- System page, Web Console 211
- System states 329
- System versus nodes 11
- Systems
  - adding with Java Console 189
  - administering with command line 101
  - deleting with Java Console 190
  - displaying node ID with command line 101
  - failure detection 288
  - freezing with Java Console 191
  - freezing with Web Console 228
  - modifying with Java Console 171
  - panic 289
  - querying with command line 92
  - states 329
  - unfreezing with Java Console 192
  - unfreezing with Web Console 229

## T

- Templates
  - adding resources 176
  - configuring service groups 172
- Troubleshooting
  - logging 293
  - notifier 321
  - resources 320
  - service groups 317
  - startup 316
  - Web Console 322
- Type-dependent attributes 38
- Type-independent attributes 38
- types.cf 30
- Type-specific attributes 39

## U

- User privileges
  - changing with Java Console 153
  - Cluster Administrator 74
  - Cluster Guest 75
  - Cluster Operator 74
  - command matrices 77
  - definition of 73

- Group Administrator 74
- Group Operator 74
- setting explicitly 75
- Utilities
  - getcomms.pl 314
  - hacf 41
  - hagetcf 315
  - halicf 85
  - hanotify 263

## V

- VCS
  - administering configuration with Java Console 196
  - closing configuration with Java Console 195
  - cluster definition 11
  - detecting application failure 5
  - detecting node failure 5
  - environment variables 83
  - event triggers 271
  - impact on performance 281
  - jeopardy 296
  - logging 293
  - modifying a configuration 193
  - network partitions 295
  - notification 259
  - opening configuration from Java Console 193
  - querying with command line 92
  - retrieving diagnostics with hagetcf 315
  - saving configuration with Java Console 194
  - scheduling classes 291
  - service group workload management 237
  - SNMP and SMTP 259
  - specifying priorities 291
  - split brain 295
  - starting on a system 85
  - starting on a system with command line 85
  - starting on single node 86
  - stopping on a system with command line 87
  - troubleshooting resources 320
  - troubleshooting service groups 317
  - troubleshooting startup 316
  - troubleshooting Web Console 322



---

VCS configuration files  
  generating 30  
  main.cf 30  
  types.cf 30  
Vector attribute dimension 38

## W

Web Console  
  All Attributes page 208  
  All Groups page 206  
  All Systems page 206  
  All Types page 207  
  autoenabling service groups 224  
  bringing resources online 226  
  bringing service groups online 221  
  clearing resources 227  
  Cluster Summary Page 205  
  configuring manually 215  
  connecting from URL 217  
  conventions 204  
  definition of 203  
  disabling service groups 223  
  editing attributes 230  
  editing Log display 233  
  enabling service groups 223  
  flushing service groups 225

freezing service groups 224  
freezing systems 228  
installation 215  
Log Details page 209  
navigating 218  
opening configuration 220  
prerequisites 217  
probing resources 227  
refresh mode 219  
resetting Log display 234  
Resource Dependency page 214  
Resource page 213  
Resource Type page 212  
saving configuration 220  
Service Group page 210  
switching service groups 222  
System page 211  
taking resources offline 226  
taking service groups offline 222  
troubleshooting 322  
unfreezing service groups 225  
unfreezing systems 229

## Z

Zoom panel 134

