

Cluster Server 7.3.1 Administrator's Guide - Windows

Last updated: 2017-11-05

Legal Notice

Copyright © 2017 Veritas Technologies LLC. All rights reserved.

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

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

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

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

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

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

Veritas Technologies LLC
500 E Middlefield Road
Mountain View, CA 94043

<http://www.veritas.com>

Technical Support

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

<https://www.veritas.com/support>

You can manage your Veritas account information at the following URL:

<https://my.veritas.com>

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

Worldwide (except Japan)

CustomerCare@veritas.com

Japan

CustomerCare_Japan@veritas.com

Documentation

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

<https://sort.veritas.com/documents>

Documentation feedback

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

doc.feedback@veritas.com

You can also see documentation information or ask a question on the Veritas community site:

<http://www.veritas.com/community/>

Veritas Services and Operations Readiness Tools (SORT)

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

https://sort.veritas.com/data/support/SORT_Data_Sheet.pdf

Contents

Section 1	Clustering concepts and terminology	
	19
Chapter 1	Introducing Cluster Server	20
	About Cluster Server	20
	How VCS detects failure	20
	How VCS ensures application availability	21
	About cluster control guidelines	22
	Defined start, stop, and monitor procedures	22
	Ability to restart the application in a known state	23
	External data storage	23
	Licensing and host name issues	24
	About the physical components of VCS	24
	About VCS nodes	24
	About shared storage	25
	About networking	25
	Logical components of VCS	25
	About resources and resource dependencies	25
	Categories of resources	27
	About resource types	28
	About service groups	28
	Types of service groups	29
	About the ClusterService group	30
	About agents in VCS	30
	About agent functions	31
	Agent classifications	33
	VCS agent framework	33
	About cluster control, communications, and membership	33
	About security services	35
	Components for administering VCS	36
	Putting the pieces together	37

Chapter 2	About cluster topologies	39
	Basic failover configurations	39
	Asymmetric or active / passive configuration	39
	Symmetric or active / active configuration	40
	About N-to-1 configuration	41
	About advanced failover configurations	42
	About the N + 1 configuration	42
	About the N-to-N configuration	43
	Cluster topologies and storage configurations	44
	About basic shared storage cluster	44
	About campus, or metropolitan, shared storage cluster	45
	About shared nothing clusters	46
	About replicated data clusters	46
	About global clusters	47
Chapter 3	VCS configuration concepts	49
	About configuring VCS	49
	VCS configuration language	50
	About the main.cf file	50
	About the SystemList attribute	53
	Initial configuration	54
	Including multiple .cf files in main.cf	54
	About the types.cf file	55
	About VCS attributes	56
	About attribute data types	56
	About attribute dimensions	56
	About attributes and cluster objects	57
	Attribute scope across systems: global and local attributes	58
	About attribute life: temporary attributes	59
	Size limitations for VCS objects	59
	VCS keywords and reserved words	60
	VCS environment variables	60
Section 2	Administration - Putting VCS to work	
	65
Chapter 4	About the VCS user privilege model	66
	About VCS user privileges and roles	66
	VCS privilege levels	66
	User roles in VCS	67

	Hierarchy in VCS roles	68
	User privileges for CLI commands	68
	User privileges for cross-cluster operations	69
	User privileges for clusters that run in secure mode	69
	How administrators assign roles to users	69
	User privileges for OS user groups for clusters running in secure mode	70
	VCS privileges for users with multiple roles	71
Chapter 5	Getting started with VCS	73
	Configuring the cluster using the Cluster Configuration Wizard	73
	Configuring notification	82
	Configuring Wide-Area Connector process for global clusters	85
	About configuring a cluster from the command line	87
	About preparing for a silent configuration	87
	Running the silent configuration utility	94
Chapter 6	Administering the cluster from the command line	96
	About administering VCS from the command line	97
	Symbols used in the VCS command syntax	97
	How VCS identifies the local system	98
	About specifying values preceded by a dash (-)	98
	About the -modify option	98
	Encrypting VCS passwords	98
	Encrypting agent passwords	99
	Starting VCS	102
	Stopping the VCS engine and related processes	103
	About stopping VCS without the -force option	104
	About stopping VCS with options other than the -force option	104
	About controlling the hastop behavior by using the EngineShutdown attribute	104
	Additional considerations for stopping VCS	105
	About managing VCS configuration files	105
	About the hacf utility	106
	About multiple versions of .cf files	106
	Verifying a configuration	106
	Scheduling automatic backups for VCS configuration files	106
	Saving a configuration	107
	Setting the configuration to read or write	107

Displaying configuration files in the correct format	108
About managing VCS users from the command line	108
Adding a user	108
Assigning and removing user privileges	110
Modifying a user	111
Deleting a user	111
Displaying a user	112
About querying VCS	112
Querying service groups	112
Querying resources	114
Querying resource types	115
Querying agents	116
Querying systems	117
Querying clusters	117
Querying status	118
Querying log data files (LDFs)	119
Using conditional statements to query VCS objects	120
About administering service groups	121
Adding and deleting service groups	121
Modifying service group attributes	122
Bringing service groups online	124
Taking service groups offline	125
Switching service groups	125
Freezing and unfreezing service groups	126
Enabling and disabling priority based failover for a service group	127
Enabling and disabling service groups	128
Clearing faulted resources in a service group	129
Linking and unlinking service groups	129
Administering agents	130
About administering resources	130
About adding resources	130
Adding resources	131
Deleting resources	131
Adding, deleting, and modifying resource attributes	131
Defining attributes as local	133
Linking and unlinking resources	135
Bringing resources online	136
Taking resources offline	136
Probing a resource	137
Clearing a resource	137
About administering resource types	138
Adding, deleting, and modifying resource types	138

Overriding resource type static attributes	139
Administering systems	140
About administering clusters	141
Retrieving version information	141
Using the -wait option in scripts that use VCS commands	142
About administering simulated clusters from the command line	142

Chapter 7

Configuring resources and applications in VCS

.....	144
About configuring resources and applications	145
Considerations for Windows Server systems	145
About Virtual Business Services	146
Features of Virtual Business Services	147
Sample virtual business service configuration	148
About Intelligent Resource Monitoring (IMF)	150
VCS changes to support IMF	151
VCS agents that support IMF	153
How IMF works	154
How to enable IMF	154
How to disable IMF	155
Recommended settings	156
About fast failover	157
VCS changes for fast failover	157
Enabling fast failover for disk groups	157
How VCS monitors storage components	158
Shared storage—if you use NetApp filers	158
Shared storage—if you use SFW to manage cluster dynamic disk groups	159
Shared storage—if you use Windows LDM to manage shared disks	159
Non-shared storage—if you use SFW to manage dynamic disk groups	160
Non-shared storage—if you use Windows LDM to manage local disks	160
Non-shared storage—if you use VMware storage	161
About storage configuration	161
About managing storage using Windows Logical Disk Manager	162
About managing storage in a Network Appliance storage environment	166
About managing shared storage using Storage Foundation for Windows	168

Managing storage	171
About configuring network resources	172
About configuring IP addresses on the systems	172
About configuring virtual computer names	174
About configuring file shares	175
Before you configure a file share service group	175
Configuring file shares using the wizard	177
Modifying a file share service group using the wizard	184
Deleting a file share service group using the wizard	185
Creating non-scoped file shares configured with VCS	185
Making non-scoped file shares accessible while using virtual server name or IP address if NetBIOS and WINS are disabled	187
About configuring IIS sites	188
Before you configure an IIS service group	189
Fixing the IPv6 address configuration for FTP sites	191
Installing IIS on Windows Server Core	191
Configuring an IIS service group using the wizard	193
Modifying an IIS service group using the wizard	197
Deleting an IIS service group using the wizard	198
About configuring services	199
About configuring a service using the GenericService agent	199
Before you configure a service using the GenericService agent	199
Configuring a service using the GenericService agent	200
About configuring a service using the ServiceMonitor agent	201
Before you configure a service using the ServiceMonitor agent	201
Configuring a service using the ServiceMonitor agent	202
About configuring processes	202
Before you configure processes	203
Configuring processes using the Process agent	203
About configuring Microsoft Message Queuing (MSMQ)	204
Before you configure the MSMQ service group	205
Configuring the MSMQ resource using the command-line utility	206
Configuring the MSMQ service group using the wizard	208
Modifying an MSMQ service group using the wizard	212
Configuring MSMQ agent to check port bindings more than once	213
Binding an MSMQ instance to the correct IP address	214
Checking whether MSMQ is listening for messages	215
About configuring the infrastructure and support agents	216
About configuring notification	217

Configuring registry replication	217
Configuring a proxy resource	220
Configuring a phantom resource	220
Configuring file resources	221
Configuring a RemoteGroup resource	221
About configuring applications using the Application Configuration Wizard	222
Before you configure service groups using the Application Configuration wizard	222
Adding resources to a service group	223
Configuring service groups using the Application Configuration Wizard	232
Modifying an application service group	234
Deleting resources from a service group	236
Deleting an application service group	236
Configuring the service group in a non-shared storage environment	237
Setting the timeout duration for which the VMNSDg agent waits for all the disks to arrive before importing the disk group	240
About the VCS Application Manager utility	241
Managing applications in virtual server context	241
About testing resource failover using virtual fire drills	243
About virtual fire drills	243
About infrastructure checks and fixes for supported agents	243
About running a virtual fire drill	244
HA and DR configurations using InfoScale in AWS	244
Failover within the same subnet of an AZ using virtual private IP	245
Failover across subnets using overlay IP	247
Public access to InfoScale cluster nodes in AWS using Elastic IP	250
DR from on-premise cluster to AWS	251
HA and DR configurations in Azure environment	256
Failover within the same subnet using a private IP	257
Failover across subnets using overlay IP	259
Public access to cluster nodes in Azure environment using public IP	261
DR across Azure Regions or VNets and from on-premises cluster to Azure	262

Chapter 8	Modifying the cluster configuration	271
	About modifying the cluster configuration	271
	Adding nodes to a cluster	272
	Removing nodes from a cluster	276
	Reconfiguring a cluster	278
	Configuring single sign-on for the cluster manually	282
	Configuring the ClusterService group	284
	Configuring notification	286
	Configuring the wide-area connector process for global clusters	287
	Deleting a cluster configuration	288
Section 3	Administration - Beyond the basics	290
Chapter 9	Controlling VCS behavior	291
	VCS behavior on resource faults	291
	Critical and non-critical resources	292
	VCS behavior diagrams	292
	About controlling VCS behavior at the service group level	294
	About the AutoRestart attribute	294
	About controlling failover on service group or system faults	294
	About defining failover policies	295
	About system zones	296
	Load-based autostart	297
	About freezing service groups	297
	About controlling Clean behavior on resource faults	297
	Clearing resources in the ADMIN_WAIT state	298
	About controlling fault propagation	300
	Customized behavior diagrams	300
	VCS behavior for resources that support the intentional offline functionality	301
	About controlling VCS behavior at the resource level	303
	Resource type attributes that control resource behavior	303
	How VCS handles resource faults	304
	VCS behavior after a resource is declared faulted	309
	About disabling resources	311
	Changing agent file paths and binaries	314
	Service group workload management	315
	About enabling service group workload management	315
	System capacity and service group load	315
	System limits and service group prerequisites	317

	About capacity and limits	317
	Sample configurations depicting workload management	317
	System and Service group definitions	318
	Sample configuration: Basic four-node cluster	319
	Sample configuration: Complex four-node cluster	323
	Sample configuration: Server consolidation	327
Chapter 10	The role of service group dependencies	335
	About service group dependencies	335
	About dependency links	335
	About dependency limitations	339
	Service group dependency configurations	339
	About failover parent / failover child	339
	Frequently asked questions about group dependencies	353
	About linking service groups	355
	VCS behavior with service group dependencies	355
	Online operations in group dependencies	355
	Offline operations in group dependencies	356
	Switch operations in group dependencies	356
Chapter 11	VCS event notification	357
	About VCS event notification	357
	Event messages and severity levels	359
	About persistent and replicated message queue	359
	How HAD deletes messages	359
	Components of VCS event notification	360
	About the notifier process	360
	About the hanotify utility	361
	About VCS events and traps	362
	Events and traps for clusters	362
	Events and traps for agents	363
	Events and traps for resources	363
	Events and traps for systems	365
	Events and traps for service groups	365
	SNMP-specific files	367
	Trap variables in VCS MIB	367
	About monitoring aggregate events	370
	How to detect service group failover	370
	How to detect service group switch	370
	About configuring notification	371

Prerequisites for wide-area heartbeats	395
Prerequisites for ClusterService group	395
Prerequisites for replication setup	395
Prerequisites for clusters running in secure mode	396
Setting up a global cluster	396
Preparing the application for the global environment	397
Configuring the ClusterService group	398
Configuring replication resources in VCS	399
Linking the application and replication service groups	401
Configuring the second cluster	401
Linking clusters	402
Configuring the Steward process (optional)	403
Stopping the Steward process	405
Configuring the global service group	405
About cluster faults	407
About the type of failure	407
Switching the service group back to the primary	407
About setting up a disaster recovery fire drill	408
About creating and configuring the fire drill service group manually	409
Multi-tiered application support using the RemoteGroup agent in a global environment	412
Test scenario for a multi-tiered environment	414
About the main.cf file for cluster 1	415
About the main.cf file for cluster 2	416
About the main.cf file for cluster 3	417
About the main.cf file for cluster 4	418

Chapter 14

Administering global clusters from Cluster Manager (Java console)	420
About global clusters	420
Adding a remote cluster	421
Deleting a remote cluster	423
Administering global service groups	426
Converting local and global groups	426
Bringing a service group online in a remote cluster	428
Taking a service group offline in a remote cluster	429
Switching a service group to a remote cluster	429
Administering global heartbeats	430
Adding a global heartbeat	430
Modifying a global heartbeat	430
Deleting a global heartbeat	431

VCS performance consideration when a service group comes online	457
VCS performance consideration when a service group goes offline	458
VCS performance consideration when a resource fails	458
VCS performance consideration when a system fails	459
VCS performance consideration when a network link fails	459
VCS performance consideration when a system panics	460
VCS performance consideration when a service group switches over	462
VCS performance consideration when a service group fails over	462
Monitoring CPU usage	463
VCS agent statistics	464
Tracking monitor cycle times	465
VCS attributes enabling agent statistics	465
About VCS performance with non-HA products	466
About VCS performance with SFW	467

Chapter 18	Troubleshooting and recovery for VCS	468
	VCS message logging	468
	VCW logs	469
	VCWsilent logs	470
	Solutions wizard logs	470
	Message catalogs	470
	Handling network failure	471
	Disabling failover	472
	Example of how VCS handles network failure	472
	Network partitioning	475
	When VCS shuts down a system	476
	Pre-existing network partitions	476
	Seeding of VCS clusters	476
	Reconnecting the private network	478
	Troubleshooting VCS startup	478
	Low Latency Transport (LLT)	479
	Group Membership Atomic Broadcast (GAB)	480
	Verifying LLT, GAB, and cluster operation	480
	VCS startup errors	487
	Cluster name contains Unicode characters	489
	Cluster ID is not unique over a network	489
	Troubleshooting secure clusters	490
	Troubleshooting service groups	491

	ClusterService group configuration	493
	Troubleshooting resources	493
	Troubleshooting notification	495
	Troubleshooting and recovery for global clusters	495
	Disaster declaration	495
	Lost heartbeats and the inquiry mechanism	496
	VCS alerts	496
	Troubleshooting the steward process	498
	VCS utilities	499
	The getcomms utility	499
	The hagetcf utility	500
	The NICTest utility	503
	The VCSRegUtil utility	504
	The havol utility	504
	The vmgetdrive utility	507
	Configuring the VCS Helper service manually	508
Section 6	Appendixes	510
Appendix A	VCS user privileges—administration matrices	511
	About administration matrices	511
	Administration matrices	511
	Agent Operations (haagent)	512
	Attribute Operations (haattr)	512
	Cluster Operations (haclus, haconf)	512
	Service group operations (hagrp)	513
	Heartbeat operations (hahb)	514
	Log operations (halog)	515
	Resource operations (hares)	515
	System operations (hasys)	516
	Resource type operations (hatype)	517
	User operations (hauser)	517
Appendix B	Cluster and system states	519
	Remote cluster states	519
	Examples of cluster state transitions	520
	System states	521
	Examples of system state transitions	523

Appendix C	VCS attributes	524
	About attributes and their definitions	524
	Resource attributes	525
	Resource type attributes	534
	Service group attributes	546
	System attributes	570
	Cluster attributes	583
	Heartbeat attributes (for global clusters)	596
	Remote cluster attributes	598
Appendix D	Configuring LLT over UDP	602
	About configuring LLT over UDP	602
	When to use LLT over UDP	602
	LLT over UDP configuration	603
	The link command in the lltab file	603
	The set-addr command in the lltab file	604
	Selecting UDP ports	604
	Sample configuration: Direct-attached links	605
	Sample configuration: Links crossing IP routers	606
	Issues and limitations	608
	VCW does not support configuring broadcasting for UDP	608
	If the network adapters are unable to ping each other, the cluster nodes may not get GAB membership	608
Appendix E	Handling concurrency violation in any-to-any configurations	610
	About handling concurrency violation	610
	Concurrency violation scenario	610
	About the vcsgensvc.vbs script	611
	Sample configuration to handle concurrency violation	612
	Notes for using scripts with the Process agent	614
Appendix F	Accessibility and VCS	615
	About accessibility in VCS	615
	Navigation and keyboard shortcuts	615
	Navigation in the Java Console	615
	Navigation in the Web console	616
	Support for accessibility settings	616
	Support for assistive technologies	617

Clustering concepts and terminology

- [Chapter 1. Introducing Cluster Server](#)
- [Chapter 2. About cluster topologies](#)
- [Chapter 3. VCS configuration concepts](#)

Introducing Cluster Server

This chapter includes the following topics:

- [About Cluster Server](#)
- [About cluster control guidelines](#)
- [About the physical components of VCS](#)
- [Logical components of VCS](#)
- [Putting the pieces together](#)

About Cluster Server

Cluster Server (VCS) connects multiple, independent systems into a management framework for increased availability. Each system, or node, runs its own operating system and cooperates at the software level to form a cluster. VCS links commodity hardware with intelligent software to provide application failover and control. When a node or a monitored application fails, other nodes can take predefined actions to take over and bring up services elsewhere in the cluster.

How VCS detects failure

VCS detects failure of an application by issuing specific commands, tests, or scripts to monitor the overall health of an application. VCS also determines the health of underlying resources by supporting the applications such as file systems and network interfaces.

VCS uses a redundant network heartbeat to differentiate between the loss of a system and the loss of communication between systems. VCS can also use SCSI3-based membership coordination and data protection for detecting failure on a node and on fencing.

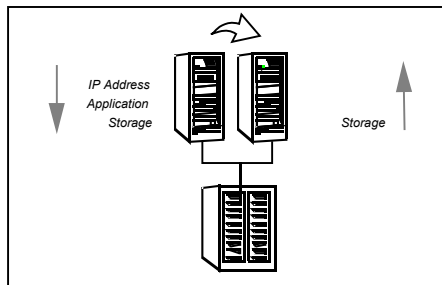
See “[About cluster control, communications, and membership](#)” on page 33.

How VCS ensures application availability

When VCS detects an application or node failure, VCS brings application services up on a different node in a cluster.

[Figure 1-1](#) shows how VCS virtualizes IP addresses and system names, so client systems continue to access the application and are unaware of which server they use.

Figure 1-1 VCS virtualizes IP addresses and system names to ensure application availability



For example, in a two-node cluster consisting of db-server1 and db-server2, a virtual address may be called db-server. Clients access db-server and are unaware of which physical server hosts the db-server.

About switchover and failover

Switchover and failover are the processes of bringing up application services on a different node in a cluster by VCS. The difference between the two processes is as follows:

Switchover	A switchover is an orderly shutdown of an application and its supporting resources on one server and a controlled startup on another server.
Failover	A failover is similar to a switchover, except the ordered shutdown of applications on the original node may not be possible due to failure of hardware or services, so the services are started on another node.

About cluster control guidelines

Most applications can be placed under cluster control provided the following guidelines are met:

- Defined start, stop, and monitor procedures
See “[Defined start, stop, and monitor procedures](#)” on page 22.
- Ability to restart in a known state
See “[Ability to restart the application in a known state](#)” on page 23.
- Ability to store required data on shared disks
See “[External data storage](#)” on page 23.
- Adherence to license requirements and host name dependencies
See “[Licensing and host name issues](#)” on page 24.

Defined start, stop, and monitor procedures

The following table describes the defined procedures for starting, stopping, and monitoring the application to be clustered:

Start procedure	<p>The application must have a command to start it and all resources it may require. VCS brings up the required resources in a specific order, then brings up the application by using the defined start procedure.</p> <p>For example, to start an Oracle database, VCS must know which Oracle utility to call, such as sqlplus. VCS must also know the Oracle user, instance ID, Oracle home directory, and the pfile.</p>
Stop procedure	<p>An individual instance of the application must be capable of being stopped without affecting other instances.</p> <p>For example, You cannot kill all httpd processes on a Web server because it also stops other Web servers.</p> <p>If VCS cannot stop an application cleanly, it may call for a more forceful method, like a kill signal. After a forced stop, a clean-up procedure may be required for various process-specific and application-specific items that may be left behind. These items include shared memory segments or semaphores.</p>

Monitor procedure The application must have a monitor procedure that determines if the specified application instance is healthy. The application must allow individual monitoring of unique instances.

For example, the monitor procedure for a Web server connects to the specified server and verifies that it serves Web pages. In a database environment, the monitoring application can connect to the database server and perform SQL commands to verify read and write access to the database.

If a test closely matches what a user does, it is more successful in discovering problems. Balance the level of monitoring by ensuring that the application is up and by minimizing monitor overhead.

Ability to restart the application in a known state

When you take an application offline, the application must close out all tasks, store data properly on shared disk, and exit. Stateful servers must not keep that state of clients in memory. States should be written to shared storage to ensure proper failover.

Commercial databases such as Oracle, Sybase, or SQL Server are good examples of well-written, crash-tolerant applications. 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 redo log file. The database confirms that the data is saved before it sends an acknowledgement to the client. After a server crashes, the database recovers to the last-known committed state by mounting the data tables and by applying the redo logs. This returns the database to the time of the crash. The client resubmits any outstanding client requests that are unacknowledged by the server, and all others are contained in the redo logs.

If an application cannot recover gracefully after a server crashes, it cannot run in a cluster environment. The takeover server cannot start up because of data corruption and other problems.

External data storage

The application must be capable of storing all required data and configuration information on shared disks. The exception to this rule is a true shared nothing cluster.

See [“About shared nothing clusters”](#) on page 46.

For example, set up SQL Server so that the binaries are installed on the local system. The shared database and configuration information reside on a shared disk.

The application must also store data to disk instead of maintaining it in memory. The takeover system must be capable of accessing all required information. This requirement precludes the use of anything inside a single system inaccessible by the peer. NVRAM accelerator boards and other disk caching mechanisms for performance are acceptable, but must be done on the external array and not on the local host.

Licensing and host name issues

The application must be capable of running on all servers that are designated as potential hosts. This requirement means strict adherence to license requirements and host name dependencies. A change of host names can lead to significant management issues when multiple systems have the same host name after an outage. To create custom scripts to modify a system host name on failover is not recommended. Veritas recommends that you configure applications and licenses to run properly on all hosts.

About the physical components of VCS

A VCS cluster comprises of systems that are connected with a dedicated communications infrastructure. VCS refers to a system that is part of a cluster as a node.

Each cluster has a unique cluster ID. Redundant cluster communication links connect systems in a cluster.

See [“About VCS nodes”](#) on page 24.

See [“About shared storage”](#) on page 25.

See [“About networking”](#) on page 25.

About VCS nodes

VCS nodes host the service groups (managed applications and their resources). Each system is connected to networking hardware, and usually to storage hardware also. The systems contain components to provide resilient management of the applications and to start and stop agents.

Nodes can be individual systems, or they can be created with domains or partitions on enterprise-class systems or on supported virtual machines. Individual cluster nodes each run their own operating system and possess their own boot device. Each node must run the same operating system within a single VCS cluster.

About shared storage

Storage is a key resource of most applications services, and therefore most service groups. You can start a managed application on a system that has access to its associated data files. Therefore, a service group can only run on all systems in the cluster if the storage is shared across all systems. In many configurations, a storage area network (SAN) provides this requirement.

See [“Cluster topologies and storage configurations”](#) on page 44.

About networking

Networking in the cluster is used for the following purposes:

- Communications between the cluster nodes and the customer systems.
- Communications between the cluster nodes.

See [“About cluster control, communications, and membership”](#) on page 33.

Logical components of VCS

VCS is comprised of several components that provide the infrastructure to cluster an application.

See [“About resources and resource dependencies”](#) on page 25.

See [“Categories of resources”](#) on page 27.

See [“About resource types”](#) on page 28.

See [“About service groups”](#) on page 28.

See [“Types of service groups”](#) on page 29.

See [“About the ClusterService group”](#) on page 30.

See [“About agents in VCS”](#) on page 30.

See [“About agent functions”](#) on page 31.

See [“VCS agent framework”](#) on page 33.

See [“About cluster control, communications, and membership”](#) on page 33.

See [“Components for administering VCS”](#) on page 36.

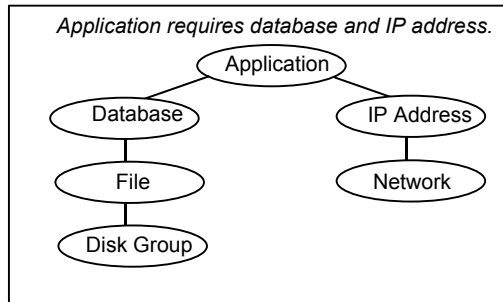
About resources and resource dependencies

Resources are hardware or software entities that make up the application. Disk groups and file systems, network interface cards (NIC), IP addresses, and applications are a few examples of resources.

Resource dependencies indicate resources that depend on each other because of application or operating system requirements. Resource dependencies are graphically depicted in a hierarchy, also called a tree, where the resources higher up (parent) depend on the resources lower down (child).

Figure 1-2 shows the hierarchy for a database application.

Figure 1-2 Sample resource dependency graph



Resource dependencies determine the order in which resources are brought online or taken offline. For example, you must import a disk group before volumes in the disk group start, and volumes must start before you mount file systems. Conversely, you must unmount file systems before volumes stop, and volumes must stop before you deport disk groups.

A parent is brought online after each child is brought online, and this continues up the tree, until finally the application starts. Conversely, to take a managed application offline, VCS stops resources by beginning at the top of the hierarchy. In this example, the application stops first, followed by the database application. Next the IP address and file systems stop concurrently. These resources do not have any resource dependency between them, and this continues down the tree.

Child resources must be brought online before parent resources are brought online. Parent resources must be taken offline before child resources are taken offline. If resources do not have parent-child interdependencies, they can be brought online or taken offline concurrently.

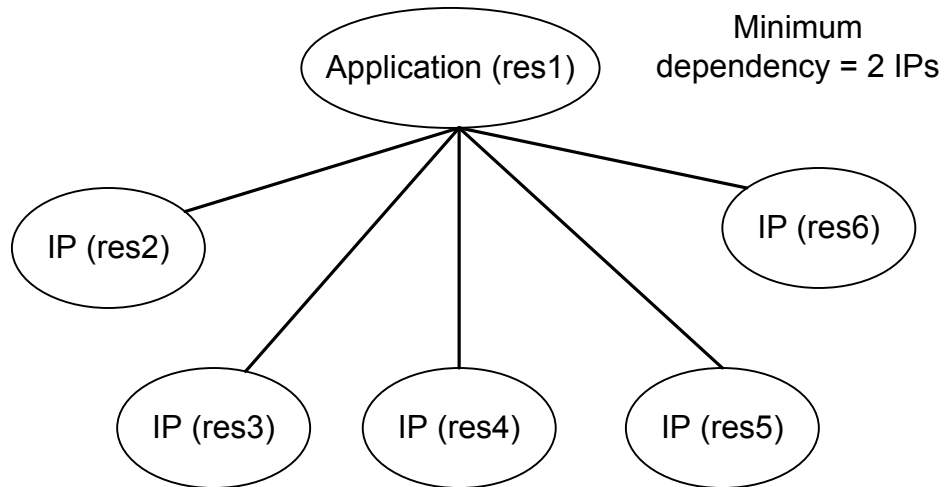
Atleast resource dependency

A new type of resource dependency has been introduced wherein a parent resource can depend on a set of child resources. The parent resource is brought online or remains online only if a minimum number of child resources in this resource set are online.

For example, if an application depends on five IPs and if this application has to be brought online or has to remain online, at least two IPs must be online. You can configure this dependency as shown in the figure below.

The system creates a set of child IP resources and the application resource will depend on this set. For this example, the assumption is that the application resource is res1 and the child IP resources are res2, res3, res4, res5, and res6.

Figure 1-3 Atleast resource dependency graph



If two or more IP resources come online, the application attempts to come online. If the number of online resources falls below the minimum requirement, (in this case: 2), resource fault is propagated up the resource dependency tree.

Note: Veritas InfoScale Operations Manager and Java GUI does not support atleast resource dependency and so the dependency is shown as normal resource dependency.

Categories of resources

Different types of resources require different levels of control.

[Table 1-1](#) describes the three categories of VCS resources.

Table 1-1 Categories of VCS resources

VCS resources	VCS behavior
On-Off	VCS starts and stops On-Off resources as required. For example, VCS imports a disk group when required, and deports it when it is no longer needed.
On-Only	VCS starts On-Only resources, but does not stop them. For example, in the case of the FileOnOnly resource, VCS creates the file. VCS does not delete the file if the service group is taken offline.
Persistent	These resources cannot be brought online or taken offline. For example, a network interface card cannot be started or stopped, but it is required to configure an IP address. A Persistent resource has an operation value of None. VCS monitors Persistent resources to ensure their status and operation. Failure of a Persistent resource triggers a service group failover.

About resource types

VCS defines a resource type for each resource it manages. For example, you can configure the NIC resource type to manage network interface cards. Similarly, you can configure an IP address using the IP resource type.

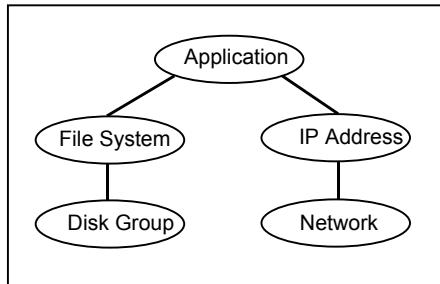
VCS includes a set of predefined resources types. For each resource type, VCS has a corresponding agent, which provides the logic to control resources.

See [“About agents in VCS”](#) on page 30.

About service groups

A service group is a virtual container that contains all the hardware and software resources that are required to run the managed application. Service groups allow VCS to control all the hardware and software resources of the managed application as a single unit. When a failover occurs, resources do not fail over individually; the entire service group fails over. If more than one service group is on a system, a group can fail over without affecting the others.

[Figure 1-4](#) shows a typical database service group.

Figure 1-4 Typical database service group

A single node can host any number of service groups, each providing a discrete service to networked clients. If the server crashes, all service groups on that node must be failed over elsewhere.

Service groups can be dependent on each other. For example, a managed application might be a finance application that is dependent on a database application. Because the managed application consists of all components that are required to provide the service, service group dependencies create more complex managed applications. When you use service group dependencies, the managed application is the entire dependency tree.

See [“About service group dependencies”](#) on page 335.

Types of service groups

VCS service groups fall in three main categories: failover, parallel, and hybrid.

About failover service groups

A failover service group runs on one system in the cluster at a time. Failover groups are used for most applications that do not support multiple systems to simultaneously access the application’s data.

About parallel service groups

A parallel service group runs simultaneously on more than one system in the cluster. A parallel service group is more complex than a failover group. Parallel service groups are appropriate for applications that manage multiple application instances that run simultaneously without data corruption.

About hybrid service groups

A hybrid service group is for replicated data clusters and is a combination of the failover and parallel service groups. It behaves as a failover group within a system zone or site and a parallel group across system zones or site.

A hybrid service group cannot fail over across system zones. VCS allows a switch operation on a hybrid group only if both systems are within the same system zone or site. If no systems exist within a zone for failover, VCS calls the nofailover trigger on the lowest numbered node.

See [“About service group dependencies”](#) on page 335.

See [“About the nofailover event trigger”](#) on page 375.

About the ClusterService group

The ClusterService group is a special purpose service group, which contains resources that are required by VCS components.

The group contains resources for the following items:

- Notification
- Wide-area connector (WAC) process, which is used in global clusters

By default, the ClusterService group can fail over to any node despite restrictions such as the node being frozen. However, if you disable the `AutoAddSystemToCSG` attribute, you can control the nodes that are included in the `SystemList`. The ClusterService group is the first service group to come online and cannot be autodisabled. The ClusterService group comes online on the first node that transitions to the running state. The VCS engine discourages the action of taking the group offline manually.

About agents in VCS

Agents are multi-threaded processes that provide the logic to manage resources. VCS has one agent per resource type. The agent monitors all resources of that type; for example, a single IP agent manages all IP resources.

When the agent starts, it obtains the necessary configuration information from VCS. It then periodically monitors the resources, and updates VCS with the resource status.

The action to bring a resource online or take it offline differs significantly for each resource type. For example, when you bring a disk group online, it requires importing the disk group. But, when you bring a database online, it requires that you start the database manager process and issue the appropriate startup commands.

VCS monitors resources when they are online and offline to ensure that they are not started on systems where they are not supposed to run. For this reason, VCS starts the agent for any resource that is configured to run on a system when the cluster is started. If no resources of a particular type are configured, the agent is not started. For example, if no Oracle resources exist in your configuration, the Oracle agent is not started on the system.

Certain agents can identify when an application has been intentionally shut down outside of VCS control. For agents that support this functionality, if an administrator intentionally shuts down an application outside of VCS control, VCS does not treat it as a fault. VCS sets the service group state as offline or partial, which depends on the state of other resources in the service group.

This feature allows administrators to stop applications that do not cause a failover. The feature is available for V51 agents. Agent versions are independent of VCS versions. For example, VCS 6.0 can run V40, V50, V51, and V52 agents for backward compatibility.

See [“VCS behavior for resources that support the intentional offline functionality”](#) on page 301.

About agent functions

Agents carry out specific functions on resources. The functions an agent performs are called entry points.

[Table 1-2](#) describes the agent functions.

Table 1-2 Agent functions

Agent functions	Role
Online	Brings a specific resource ONLINE from an OFFLINE state.
Offline	Takes a resource from an ONLINE state to an OFFLINE state.

Table 1-2 Agent functions (*continued*)

Agent functions	Role
Monitor	<p>Tests the status of a resource to determine if the resource is online or offline.</p> <p>The function runs at the following times:</p> <ul style="list-style-type: none"> ■ During initial node startup, to probe and determine the status of all resources on the system. ■ After every online and offline operation. ■ Periodically, to verify that the resource remains in its correct state. Under normal circumstances, the monitor entry point is run every 60 seconds when a resource is online. The entry point is run every 300 seconds when a resource is expected to be offline. ■ When you probe a resource using the following command: <pre># hares -probe res_name -sys system_name.</pre>
Clean	<p>Cleans up after a resource fails to come online, fails to go offline, or fails to detect as ONLINE when resource is in an ONLINE state. The clean entry point is designed to clean up after an application fails. The function ensures that the host system is returned to a valid state. For example, the clean function may remove shared memory segments or IPC resources that are left behind by a database.</p>
Action	<p>Performs actions that can be completed in a short time and which are outside the scope of traditional activities such as online and offline. Some agents have predefined action scripts that you can run by invoking the action function.</p>
Info	<p>Retrieves specific information for an online resource.</p> <p>The retrieved information is stored in the resource attribute ResourceInfo. This function is invoked periodically by the agent framework when the resource type attribute InfoInterval is set to a non-zero value. The InfoInterval attribute indicates the period after which the info function must be invoked. For example, the Mount agent may use this function to indicate the space available on the file system.</p> <p>To see the updated information, you can invoke the info agent function explicitly from the command line interface by running the following command:</p> <pre>hares -refreshinfo res [-sys system] -clus cluster -localclus</pre>

Agent classifications

The different kinds of agents that work with VCS include bundled agents, enterprise agents, and custom agents.

About bundled agents

Bundled agents are packaged with VCS. They include agents for Disk, Mount, IP, and various other resource types.

About enterprise agents

Enterprise agents control third party applications. These include agents for Oracle, Sybase, and DB2.

See the following documentation for more information:

- *Cluster Server Agent for Oracle Installation and Configuration Guide*
- *Cluster Server Agent for Sybase Installation and Configuration Guide*
- *Cluster Server Agent for DB2 Installation and Configuration Guide*

About custom agents

Custom agents are agents that customers or Veritas consultants develop. Typically, agents are developed because the user requires control of an application that the current bundled or enterprise agents do not support.

See the *Cluster Server Agent Developer's Guide*.

VCS agent framework

The VCS agent framework is a set of common, predefined functions that are compiled into each agent. These functions include the ability to connect to the Veritas High Availability Engine (HAD) and to understand common configuration attributes. The agent framework frees the developer from developing functions for the cluster; the developer instead can focus on controlling a specific resource type.

About cluster control, communications, and membership

Cluster communications ensure that 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 have joined or left the cluster, and which have failed.

See [“About the high availability daemon \(HAD\)”](#) on page 34.

See [“About Group Membership Services and Atomic Broadcast \(GAB\)”](#) on page 34.

See [“About Low Latency Transport \(LLT\)”](#) on page 35.

About the high availability daemon (HAD)

The VCS high availability daemon (HAD) runs on each system.

Also known as the Veritas High Availability Engine, HAD is responsible for the following functions:

- Builds the running cluster configuration from the configuration files
- Distributes the information when new nodes join the cluster
- Responds to operator input
- Takes corrective action when something fails.

The engine uses agents to monitor and manage resources. It collects information about resource states from the agents on the local system and forwards it to all cluster members.

The local engine also receives information from the other cluster members to update its view of the cluster. HAD operates as a replicated state machine (RSM). The engine that runs on each node has a completely synchronized view of the resource status on each node. Each instance of HAD follows the same code path for corrective action, as required.

The RSM is maintained through the use of a purpose-built communications package. The communications package consists of the protocols Low Latency Transport (LLT) and Group Membership Services and Atomic Broadcast (GAB).

The hashadow process monitors HAD and restarts it when required.

About Group Membership Services and Atomic Broadcast (GAB)

The Group Membership Services and Atomic Broadcast protocol (GAB) is responsible for the following cluster membership and cluster communications functions:

- Cluster Membership
GAB maintains cluster membership by receiving input on the status of the heartbeat from each node by LLT. When a system no longer receives heartbeats from a peer, it marks the peer as DOWN and excludes the peer from the cluster. In VCS, memberships are sets of systems participating in the cluster.

VCS has the following types of membership:

- A regular membership includes systems that communicate with each other across more than one network channel.
- A jeopardy membership includes systems that have only one private communication link.
- A visible membership includes systems that have GAB running but the GAB client is no longer registered with GAB.
- Cluster Communications
GAB's second function is reliable cluster communications. GAB provides guaranteed delivery of point-to-point and broadcast messages to all nodes. The Veritas High Availability Engine uses a private IOCTL (provided by GAB) to tell GAB that it is alive.

About Low Latency Transport (LLT)

VCS uses private network communications between cluster nodes for cluster maintenance. The Low Latency Transport functions as a high-performance, low-latency replacement for the IP stack, and is used for all cluster communications. Veritas recommends at least two independent networks between all cluster nodes.

For detailed information on the setting up networks, see *Cluster Server Configuration and Upgrade Guide*.

These networks provide the required redundancy in the communication path and enable VCS to differentiate between a network failure and a system failure.

LLT has the following two major functions:

- Traffic distribution
LLT distributes (load balances) internode communication across all available private network links. This distribution means that all cluster communications are evenly distributed across all private network links (maximum eight) for performance and fault resilience. If a link fails, traffic is redirected to the remaining links.
- Heartbeat
LLT is responsible for sending and receiving heartbeat traffic over network links. The Group Membership Services function of GAB uses this heartbeat to determine cluster membership.

About security services

VCS uses the VCS Authentication Service to provide secure communication between cluster nodes. VCS uses digital certificates for authentication and uses SSL to encrypt communication over the public network.

In secure mode:

- VCS uses platform-based authentication.
- VCS does not store user passwords.
- All VCS users are system and domain users and are configured using fully-qualified user names. For example, `administrator@vcsdomain`. VCS provides a single sign-on mechanism, so authenticated users do not need to sign on each time to connect to a cluster.

For secure communication, VCS components acquire credentials from the authentication broker that is configured on the local system. In VCS 6.0 and later, a root and authentication broker is automatically deployed on each node when a secure cluster is configured. The acquired certificate is used during authentication and is presented to clients for the SSL handshake.

VCS and its components specify the account name and the domain in the following format:

- **HAD Account**

```
name = HAD
domain = VCS_SERVICES@Cluster UUID
```

- **CmdServer**

```
name = CMDSERVER
domain = VCS_SERVICES@Cluster UUID
```

- **Wide-area connector**

```
name = _WAC_GCO_(systemname)
domain = HA_SERVICES@(fully_qualified_system_name)
```

Components for administering VCS

VCS provides several components to administer clusters.

[Table 1-3](#) describes the components that VCS provides to administer clusters:

Table 1-3 VCS components to administer clusters

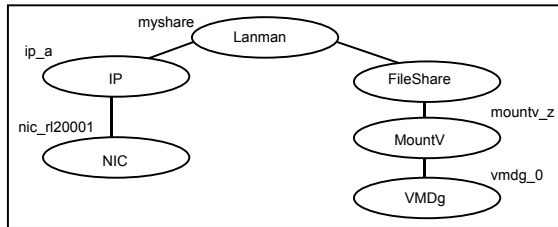
VCS components	Description
Veritas InfoScale Operations Manager	<p>A Web-based graphical user interface for monitoring and administering the cluster.</p> <p>Install the Veritas InfoScale Operations Manager on a management server outside the cluster to manage multiple clusters.</p> <p>See the Veritas InfoScale Operations Manager documentation for more information.</p>
Cluster Manager (Java Console)	<p>A cross-platform Java-based graphical user interface that provides complete administration capabilities for your cluster. The console runs on any system inside or outside the cluster, on any operating system that supports Java. You cannot use Java Console if the external communication port for VCS is not open. By default, the external communication port for VCS is 14141.</p> <p>Note: Veritas recommends Veritas InfoScale Operations Manager be used as the graphical user interface to monitor and administer the cluster. Cluster Manager (Java Console) may not support features released in VCS 6.0 and later versions.</p>
VCS command line interface (CLI)	<p>The VCS command-line interface provides a comprehensive set of commands for managing and administering the cluster.</p> <p>See “About administering VCS from the command line” on page 97.</p>

Putting the pieces together

In the following example, a two-node cluster shares directories to clients. Both nodes are connected to shared storage, which enables them access to the directories that are being shared. A single service group, "FileShare_Group," is configured to fail over between System A and System B. The service group consists of various resources, each with a different resource type.

The Veritas High Availability Engine (HAD) reads the configuration file, determines what agents are required to control the resources in the service group, and starts the agents. HAD uses resource dependencies to determine the order in which to bring the resources online. VCS issues online commands to the corresponding agents in the correct order.

[Figure 1-5](#) shows the dependency graph for the service group FileShare_Group.

Figure 1-5 Dependency graph for the service group FileShare_Group

In this configuration, HAD starts agents for the disk group, mount, share, NIC, and IP on all systems configured to run FileShare_Group.

The resource dependencies are configured as follows:

- The MountV resource requires that the VMDg resource is online before you bring it online. The FileShare resource requires that the MountV resource is online before you bring it online.
- The IP resource requires that the NIC resource is online before you bring it online. The NIC resource is a persistent resource and does not need to be started.
- The Lanman resource requires that the FileShare and IP resources are online before you can bring them online.

You can configure the service group to start automatically on either node in the preceding example. It then can move or fail over to the second node on command or automatically if the first node fails. On failover or relocation, to make the resources offline on the first node, VCS begins at the top of the graph. When it starts them on the second node, it begins at the bottom.

About cluster topologies

This chapter includes the following topics:

- [Basic failover configurations](#)
- [About advanced failover configurations](#)
- [Cluster topologies and storage configurations](#)

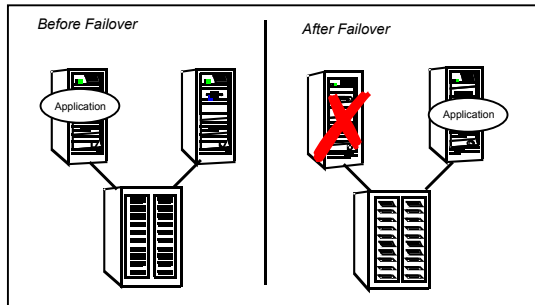
Basic failover configurations

The basic failover configurations include asymmetric, symmetric, and N-to-1.

Asymmetric or active / passive configuration

In an asymmetric configuration, an application runs on a primary server. A dedicated redundant server is present to take over on any failure. The redundant server is not configured to perform any other functions.

[Figure 2-1](#) shows failover within an asymmetric cluster configuration, where a database application is moved, or failed over, from the master to the redundant server.

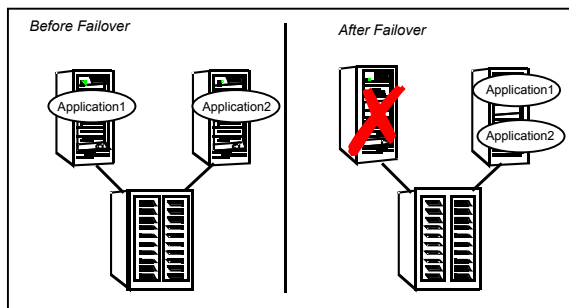
Figure 2-1 Asymmetric failover

This configuration is the simplest and most reliable. The redundant server is on stand-by with full performance capability. If other applications are running, they present no compatibility issues.

Symmetric or active / active configuration

In a symmetric configuration, each server is configured to run a specific application or service and provide redundancy for its peer. In this example, each server runs one application service group. When a failure occurs, the surviving server hosts both application groups.

Figure 2-2 shows failover within a symmetric cluster configuration.

Figure 2-2 Symmetric failover

Symmetric configurations appear more efficient in terms of hardware utilization. In the asymmetric example, the redundant server requires only as much processor power as its peer. On failover, performance remains the same. In the symmetric example, the redundant server requires adequate processor power to run the existing application and the new application it takes over.

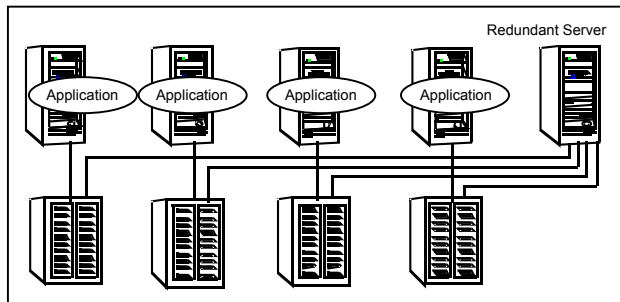
Further issues can arise in symmetric configurations when multiple applications that run on the same system do not co-exist properly. Some applications work well with multiple copies started on the same system, but others fail. Issues also can arise when two applications with different I/O and memory requirements run on the same system.

About N-to-1 configuration

An N-to-1 failover configuration reduces the cost of hardware redundancy and still provides a potential, dedicated spare. In an asymmetric configuration no performance penalty exists. No issues exist with multiple applications running on the same system; however, the drawback is the 100 percent redundancy cost at the server level.

Figure 2-3 shows an N to 1 failover configuration.

Figure 2-3 N-to-1 configuration

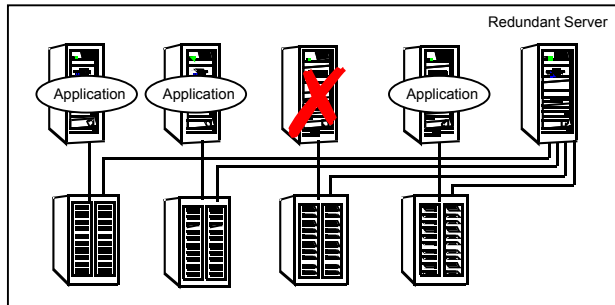


An N-to-1 configuration is based on the concept that multiple, simultaneous server failures are unlikely; therefore, a single redundant server can protect multiple active servers. When a server fails, its applications move to the redundant server. For example, in a 4-to-1 configuration, one server can protect four servers. This configuration reduces redundancy cost at the server level from 100 percent to 25 percent. In this configuration, a dedicated, redundant server is cabled to all storage and acts as a spare when a failure occurs.

The problem with this design is the issue of failback. When the failed server is repaired, you must manually fail back all services that are hosted on the failover server to the original server. The failback action frees the spare server and restores redundancy to the cluster.

Figure 2-4 shows an N to 1 failover requiring failback.

Figure 2-4 N-to-1 failover requiring failback



Most shortcomings of early N-to-1 cluster configurations are caused by the limitations of storage architecture. Typically, it is impossible to connect more than two hosts to a storage array without complex cabling schemes and their inherent reliability problems, or expensive arrays with multiple controller ports.

About advanced failover configurations

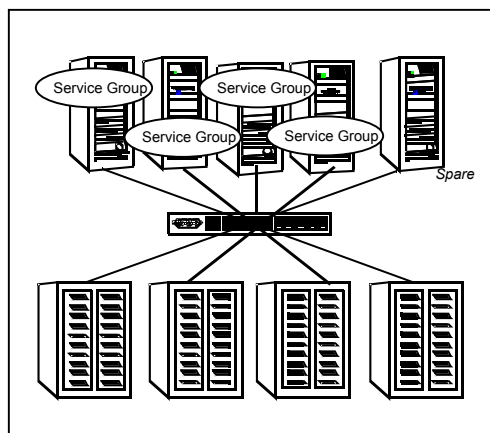
Advanced failover configuration for VCS include N + 1 and N-to-N configurations.

About the N + 1 configuration

With the capabilities introduced by storage area networks (SANs), you cannot only create larger clusters, you can also connect multiple servers to the same storage.

Figure 2-5 shows an N+1 cluster failover configuration.

Figure 2-5 N+1 configuration

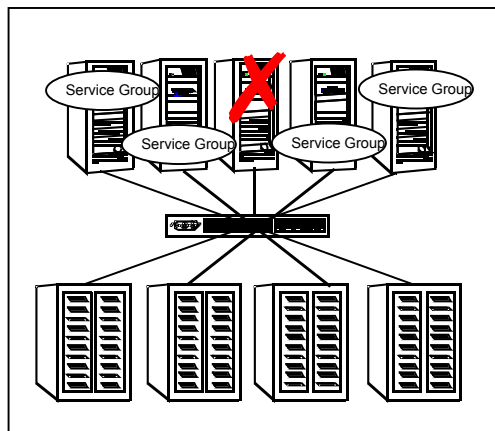


A dedicated, redundant server is no longer required in the configuration. Instead of N-to-1 configurations, you can use an N+1 configuration. In advanced N+1 configurations, an extra server in the cluster is spare capacity only.

When a server fails, the application service group restarts on the spare. After the server is repaired, it becomes the spare. This configuration eliminates the need for a second application failure to fail back the service group to the primary system. Any server can provide redundancy to any other server.

Figure 2-6 shows an N+1 cluster failover configuration requiring failback.

Figure 2-6 N+1 cluster failover configuration requiring failback

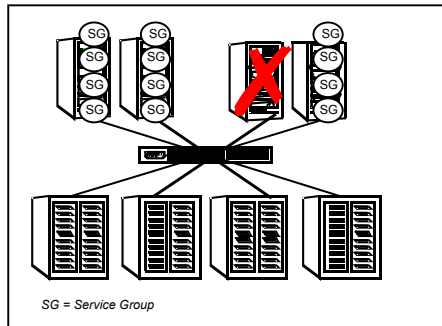


About the N-to-N configuration

An N-to-N configuration refers to multiple service groups that run on multiple servers, with each service group capable of being failed over to different servers. For example, consider a four-node cluster in which each node supports three critical database instances.

Figure 2-7 shows an N to N cluster failover configuration.

Figure 2-7 N-to-N configuration



If any node fails, each instance is started on a different node. this action ensures that no single node becomes overloaded. This configuration is a logical evolution of $N + 1$; it provides cluster standby capacity instead of a standby server.

N-to-N configurations require careful testing to ensure that all applications are compatible. You must specify a list of systems on which a service group is allowed to run in the event of a failure.

Cluster topologies and storage configurations

The commonly-used cluster topologies include the following:

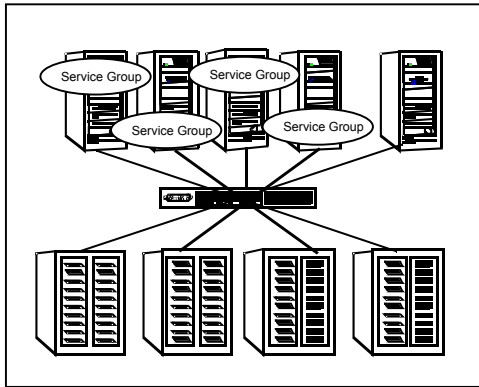
- Shared storage clusters
- Campus clusters
- Shared nothing clusters
- Replicated data clusters
- Global clusters

About basic shared storage cluster

In this configuration, a single cluster shares access to a storage device, typically over a SAN. You can only start an application on a node with access to the required storage. For example, in a multi-node cluster, any node that is designated to run a specific database instance must have access to the storage where the database's tablespaces, redo logs, and control files are stored. Such a shared disk architecture is also the easiest to implement and maintain. When a node or application fails, all data that is required to restart the application on another node is stored on the shared disk.

[Figure 2-8](#) shows a shared disk architecture for a basic cluster.

Figure 2-8 Shared disk architecture for basic cluster

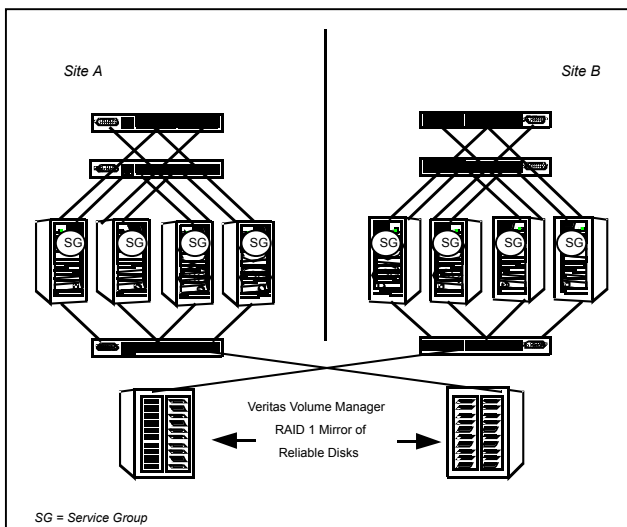


About campus, or metropolitan, shared storage cluster

In a campus environment, you use VCS and Veritas Volume Manager to create a cluster that spans multiple datacenters or buildings. Instead of a single storage array, data is mirrored between arrays by using Veritas Volume Manager. This configuration provides synchronized copies of data at both sites. This procedure is identical to mirroring between two arrays in a datacenter; only now it is spread over a distance.

Figure 2-9 shows a campus shared storage cluster.

Figure 2-9 Campus shared storage cluster



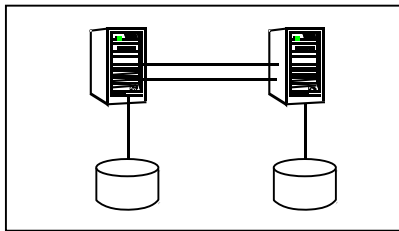
A campus cluster requires two independent network links for heartbeat, two storage arrays each providing highly available disks, and public network connectivity between buildings on same IP subnet. If the campus cluster setup resides on different subnets with one for each site, then use the VCS Lanman agent to handle the network changes or issue the DNS changes manually.

About shared nothing clusters

Systems in shared nothing clusters do not share access to disks; they maintain separate copies of data. VCS shared nothing clusters typically have read-only data stored locally on both systems. For example, a pair of systems in a cluster that includes a critical Web server, which provides access to a backend database. The Web server runs on local disks and does not require data sharing at the Web server level.

Figure 2-10 shows a shared nothing cluster.

Figure 2-10 Shared nothing cluster

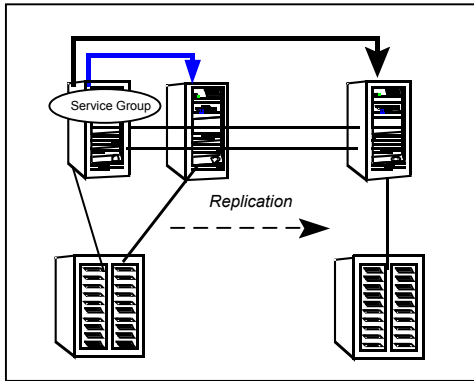


About replicated data clusters

In a replicated data cluster no shared disks exist. Instead, a data replication product synchronizes copies of data between nodes or sites. Replication can take place at the application, host, and storage levels. Application-level replication products, such as Oracle DataGuard, maintain consistent copies of data between systems at the SQL or database levels. Host-based replication products, such as Veritas Volume Replicator, maintain consistent storage at the logical volume level. Storage-based or array-based replication maintains consistent copies of data at the disk or RAID LUN level.

Figure 2-11 shows a hybrid shared storage and replicated data cluster, in which different failover priorities are assigned to nodes according to particular service groups.

Figure 2-11 Shared storage replicated data cluster



You can also configure replicated data clusters without the ability to fail over locally, but this configuration is not recommended.

See “[How VCS replicated data clusters work](#)” on page 448.

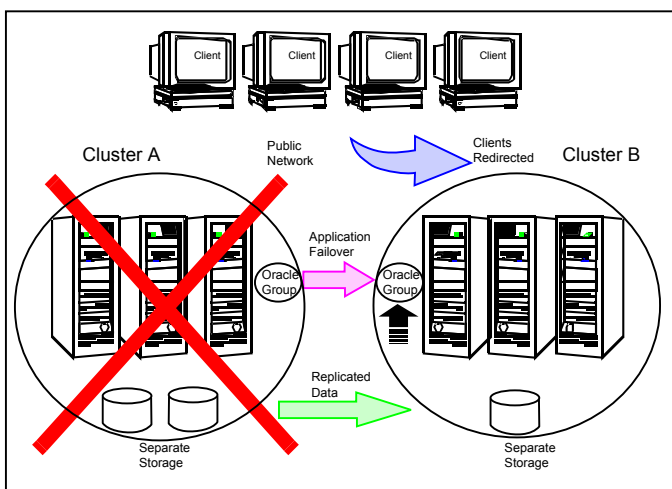
About global clusters

A global cluster links clusters at separate locations and enables wide-area failover and disaster recovery.

Local clustering provides local failover for each site or building. Campus and replicated cluster configurations offer protection against disasters that affect limited geographic regions. Large scale disasters such as major floods, hurricanes, and earthquakes can cause outages for an entire city or region. In such situations, you can ensure data availability by migrating applications to sites located considerable distances apart.

[Figure 2-12](#) shows a global cluster configuration.

Figure 2-12 Global cluster



In a global cluster, if an application or a system fails, the application is migrated to another system within the same cluster. If the entire cluster fails, the application is migrated to a system in another cluster. Clustering on a global level also requires the replication of shared data to the remote site.

VCS configuration concepts

This chapter includes the following topics:

- [About configuring VCS](#)
- [VCS configuration language](#)
- [About the main.cf file](#)
- [About the types.cf file](#)
- [About VCS attributes](#)
- [VCS keywords and reserved words](#)
- [VCS environment variables](#)

About configuring VCS

When you configure VCS, you convey to the Veritas High Availability Engine the definitions of the cluster, service groups, resources, and dependencies among service groups and resources.

VCS uses the following two configuration files in a default configuration:

- `main.cf`
Defines the cluster, including services groups and resources.
- `types.cf`
Defines the resource types.

By default, both files reside in the following directory:

```
%VCS_HOME%\conf\config
```

Additional files that are similar to `types.cf` may be present if you enabled agents such as Oracle `types.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 that are brought online after the first system derive their information from systems that are in the cluster.

You must stop the cluster if you need to modify the files manually. Changes made by editing the configuration files take effect when the cluster is restarted. The node where you made the changes should be the first node to be brought back online.

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, whose names contain the suffix `.cf`.

Several ways to generate configuration files are as follows:

- Use the Web-based Veritas InfoScale Operations Manager.
- Use Cluster Manager (Java Console).
- Use the command-line interface.
- If VCS is not running, use a text editor to create and modify the files.
- Use the VCS simulator on a Windows system to create the files.

About 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.

[Table 3-1](#) describes some of the components of the `main.cf` file:

Table 3-1 Components of the main.cf file

Components of main.cf file	Description
Include clauses	<p>Include clauses incorporate additional configuration files into main.cf. These additional files typically contain type definitions, including the types.cf file. Typically, custom agents add type definitions in their own files.</p> <pre>include "types.cf"</pre> <p>See "Including multiple .cf files in main.cf" on page 54.</p>
Cluster definition	<p>Defines the attributes of the cluster, the cluster name and the names of the cluster users.</p> <pre>cluster demo (UserNames = { admin = cDRpdxPmHzpS })</pre> <p>See "Cluster attributes" on page 583.</p>
System definition	<p>Lists the systems designated as part of the cluster. The system names must match the name returned by the command <code>uname -a</code>.</p> <p>Each service group can be configured to run on a subset of systems defined in this section.</p> <pre>system Server1 system Server2</pre> <p>See System attributes on page 570.</p>
Service group definition	<p>Service group definitions in main.cf comprise the attributes of a particular service group.</p> <pre>group FileShare_Group (SystemList = { SystemA, SystemB } AutoStartList = { SystemA })</pre> <p>See "Service group attributes" on page 546.</p> <p>See "About the SystemList attribute" on page 53.</p>

Table 3-1 Components of the main.cf file (continued)

Components of main.cf file	Description
Resource definition	<p>Defines each resource that is used in a particular service group. You can add resources in any order. The utility hacf arranges the resources alphabetically the first time the configuration file is run.</p> <pre> NIC NIC_resource (MACAddress @ system1= "02-B0-D0-D1-88-0E" MACAddress @ system2= "50-B0-D0-D1-88-23")</pre>
Resource dependency clause	<p>Defines a relationship between resources. A dependency is indicated by the keyword <code>requires</code> between two resource names.</p> <pre> IP_resource requires NIC_resource</pre> <p>In an <code>atleast</code> resource dependency, the parent resource depends on a set of child resources and a minimum number of resources from this resource set are required for the parent resource to be brought online or to remain online.</p> <pre> res1 requires atleast 2 from res2, res3,res4,res5,res6</pre> <p>The above dependency states that <code>res1</code> can be brought online or can remain online only when 2 resources from the resource set (<code>res2</code>, <code>res3</code>, <code>res4</code>, <code>res5</code>, <code>res6</code>) are online. The minimum number of resources required by the parent resource should be greater than or equal to 1 and less than the total number of child resources in the resource set.</p> <p>See “About resources and resource dependencies” on page 25.</p>

Table 3-1 Components of the main.cf file (*continued*)

Components of main.cf file	Description
Service group dependency clause	<p>To configure a service group dependency, place the keyword <code>requires</code> in the service group declaration of the main.cf file. Position the dependency clause before the resource dependency specifications and after the resource declarations.</p> <pre>requires group_x <dependency category> <dependency location> <dependency rigidity></pre> <p>See “About service group dependencies” on page 335.</p>

Note: Sample configurations for components of global clusters are listed separately.

See [“VCS global clusters: The building blocks”](#) on page 388.

About the SystemList attribute

The SystemList attribute designates all systems where a service group can come online. By default, the order of systems in the list defines the priority of systems that are used in a failover. For example, the following definition configures SystemA to be the first choice on failover, followed by SystemB, and then by SystemC.

```
SystemList = { SystemA, SystemB, SystemC }
```

You can assign system priority explicitly in the SystemList attribute by assigning numeric values to each system name. For example:

```
SystemList = { SystemA = 0, SystemB = 1, SystemC = 2 }
```

If you do not assign numeric priority values, VCS assigns a priority to the system without a number by adding 1 to the priority of the preceding system. For example, if the SystemList is defined as follows, VCS assigns the values SystemA = 0, SystemB = 2, SystemC = 3.

```
SystemList = { SystemA, SystemB = 2, SystemC }
```

Note that a duplicate numeric priority value may be assigned in some situations:

```
SystemList = { SystemA, SystemB=0, SystemC }
```

The numeric values assigned are SystemA = 0, SystemB = 0, SystemC = 1.

To avoid this situation, do not assign any numbers or assign different numbers to each system in SystemList.

Initial configuration

When VCS is installed, a basic main.cf configuration file is created with the cluster name, systems in the cluster, and a Cluster Manager user named *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 (
)
```

Including multiple .cf files in main.cf

You may choose include several configuration files in the main.cf file. For example:

```
include "applicationtypes.cf"
include "listofsystems.cf"
include "applicationgroup.cf"
```

If you include other .cf files in main.cf, the following considerations apply:

- Resource type definitions must appear before the definitions of any groups that use the resource types.

In the following example, the applicationgroup.cf file includes the service group definition for an application. The service group includes resources whose resource types are defined in the file applicationtypes.cf. In this situation, the applicationtypes.cf file must appear first in the main.cf file.

For example:

```
include "applicationtypes.cf"
include "applicationgroup.cf"
```

- If you define heartbeats outside of the main.cf file and include the heartbeat definition file, saving the main.cf file results in the heartbeat definitions getting added directly to the main.cf file.

About 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 types definition performs the following two important functions:

- Defines the type of values that may be set for each attribute.
In the following IP example, the Address attribute is classified as str, or string. See [“About attribute data types”](#) on page 56.
- Defines the parameters that are passed to the VCS engine through the ArgList attribute. The line static str ArgList[] = { xxx, yyy, zzz } defines the order in which parameters are passed to the agents for starting, stopping, and monitoring resources.

The types.cf file describes standard resource types to the VCS engine; specifically, the data required to control a specific resource.

```
type IP (
    static il8nstr ArgList[] = { Address, SubNetMask,
                               MACAddress}
    str Address
    str SubNetMask
    str MACAddress
)
```

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

- The high-availability address is configured on the interface defined by the Address attribute.
- The IP address is enclosed in double quotes because the string contains periods. See [“About attribute data types”](#) on page 56.
- 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 that it requires. All resource names must be unique in a VCS cluster.

main.cf for Windows:

```
IP IP_resource (
    Address = "192.168.1.201"
    SubNetMask = "255.255.254.0"
    MACAddress @ system1= "02-B0-D5-D1-88-0E"
    MACAddress @ system2= "04-B0-D0-D1-88-43"
)
```

types.cf for Windows:

```
type IP (  
    static il8nstr ArgList[] = { Address, SubNetMask,  
    MACAddress}  
    str Address  
    str SubNetMask  
    str MACAddress  
)
```

About VCS attributes

VCS components are configured by using attributes. Attributes contain data about the cluster, systems, service groups, resources, resource types, agent, and heartbeats if you use global clusters. 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. Attributes also have default values assigned when a value is not specified.

About attribute data types

VCS supports the following data types for attributes:

String	<p>A string is a sequence of characters that is enclosed by double quotes. A string can also contain double quotes, but the quotes must be immediately preceded by a backslash. A backslash is represented in a string as \\. Quotes are not required if a string begins with a letter, and contains only letters, numbers, dashes (-), and underscores (_).</p> <p>VCS also supports UTF-8 encoded values for some attributes.</p>
Integer	<p>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.</p>
Boolean	<p>A boolean is an integer, the possible values of which are 0 (false) and 1 (true).</p>

About attribute dimensions

VCS attributes have the following dimensions:

Scalar	A scalar has only one value. This is the default dimension.
--------	---

Vector	<p>A vector is an ordered list of values. Each value is indexed by using a positive integer beginning with zero. Use a comma (,) or a semi-colon (;) to separate values. A set of brackets ([]) after the attribute name denotes that the dimension is a vector.</p> <p>For example, an agent's ArgList is defined as:</p> <pre>static str ArgList[] = { Address, SubNetMask, MACAddress }</pre>
Keylist	<p>A keylist is an unordered list of strings, and each string is unique within the list. Use a comma (,) or a semi-colon (;) to separate values.</p> <p>For example, to designate the list of systems on which a service group will be started with VCS (usually at system boot):</p> <pre>AutoStartList = {SystemA; SystemB; SystemC}</pre>
Association	<p>An association is an unordered list of name-value pairs. Use a comma (,) or a semi-colon (;) to separate values.</p> <p>A set of braces ({}) after the attribute name denotes that an attribute is an association.</p> <p>For example, to associate the average time and timestamp values with an attribute:</p> <pre>str MonitorTimeStats{} = { Avg = "0", TS = "" }</pre>

About attributes and cluster objects

VCS has the following types of attributes, depending on the cluster object the attribute applies to:

Cluster attributes	<p>Attributes that define the cluster.</p> <p>For example, ClusterName and ClusterAddress.</p>
Service group attributes	<p>Attributes that define a service group in the cluster.</p> <p>For example, Administrators and ClusterList.</p>
System attributes	<p>Attributes that define the system in the cluster.</p> <p>For example, Capacity and Limits.</p>

Resource type attributes	<p>Attributes that define the resource types in VCS.</p> <p>These resource type attributes can be further classified as:</p> <ul style="list-style-type: none"> ■ Type-independent Attributes that all agents (or resource types) understand. Examples: RestartLimit and MonitorInterval; these can be set for any resource type. Typically, these attributes are set for all resources of a specific type. For example, setting MonitorInterval for the IP resource type affects all IP resources. ■ Type-dependent Attributes that apply to a particular resource type. These attributes appear in the type definition file (types.cf) for the agent. Example: The Address attribute applies only to the IP resource type. Attributes defined in the file types.cf apply to all resources of a particular resource type. Defining these attributes in the main.cf file overrides the values in the types.cf file for a specific resource. For example, if you set StartVolumes = 1 for the DiskGroup types.cf, it sets StartVolumes to True for all DiskGroup resources, by default. If you set the value in main.cf , it overrides the value on a per-resource basis. ■ Static These attributes apply for every resource of a particular type. These attributes are prefixed with the term static and are not included in the resource's argument list. You can override some static attributes and assign them resource-specific values. <p>See “Overriding resource type static attributes” on page 139.</p>
Resource attributes	<p>Attributes that define a specific resource.</p> <p>Some of these attributes are type-independent. For example, you can configure the Critical attribute for any resource.</p> <p>Some resource attributes are type-dependent. For example, the Address attribute defines the IP address that is associated with the IP resource. These attributes are defined in the main.cf file.</p>
Site attributes	<p>Attributes that define a site.</p> <p>For example, Preference and SystemList.</p>

Attribute scope across systems: global and local 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 IP resource type where Mac addresses and routing options are assigned per machine.

```
IP IP_resource (  
    Address = "192.168.1.201"  
    SubNetMask = "255.255.254.0"  
    MACAddress @ system1= "02-B1-D5-D1-88-0E"  
    MACAddress @ system2= "04-B0-D0-D1-88-43"  
)
```

About attribute life: temporary attributes

You can define temporary attributes in the types.cf file. The values of temporary attributes remain in memory as long as the VCS engine (HAD) is running. Values of temporary attributes are not available when HAD is restarted. These attribute values are not stored in the main.cf file.

You cannot convert temporary attributes to permanent attributes and vice-versa. When you save a configuration, VCS saves temporary attributes and their default values in the file types.cf.

The scope of these attributes can be local to a node or global across all nodes in the cluster. You can define local attributes even when the node is not part of a cluster.

You can define and modify these attributes only while VCS is running.

See [“Adding, deleting, and modifying resource attributes”](#) on page 131.

Size limitations for VCS objects

The following VCS objects are restricted to 1024 characters.

- Service group names
- Resource names
- Resource type names
- User names
- Attribute names

VCS passwords are restricted to 255 characters. You can enter a password of maximum 255 characters.

VCS keywords and reserved words

Following is a list of VCS keywords and reserved words. Note that they are case-sensitive.

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

VCS environment variables

Table 3-2 VCS environment variables

Environment Variable	Definition and Default Value
PERL5LIB	Root directory for Perl executables. (applicable only for Windows) Default: Install Drive:\Program Files\VERITAS\cluster server\lib\perl5.
VCS_CONF	Root directory for VCS configuration files. Default: Install Drive:\Program Files\VERITAS\cluster server\conf\config Note: You cannot modify this variable.
VCS_CONN_INIT_QUOTA	Maximum number of simultaneous connections accepted by the VCS engine (HAD) per client host that has not completed the handshake.
VCS_CONN_HANDSHAKE_TIMEOUT	Timeout in seconds after which the VCS engine (HAD) closes the client connection that has not completed the handshake.

Table 3-2 VCS environment variables (*continued*)

Environment Variable	Definition and Default Value
VCS_DEBUG_LOG_TAGS	<p>Enables debug logs for the VCS engine, VCS agents, and HA commands. You must set VCS_DEBUG_LOG_TAGS before you start HAD or before you execute HA commands.</p> <p>To make sure that the debug logging persists even after HAD or the cluster node is restarted, set this environment variable as follows:</p> <pre>VCS_DEBUG_LOG_TAGS="DBG_POLICY DBG_IPM DBG_TRACE DBG_INTERNAL DBG_SECURITY DBG_AGTRACE DBG_AGINFO DBG_AGDEBUG"</pre> <p>Alternatively, run the following command to add these debug logs:</p> <pre>halog -addtags DBG_POLICY DBG_IPM DBG_TRACE DBG_INTERNAL DBG_SECURITY DBG_AGTRACE DBG_AGINFO DBG_AGDEBUG</pre> <p>Note that the debug logs added using the CLI do not persist if HAD or the cluster node is restarted.</p>
VCS_DOMAIN	<p>The Security domain in which users are configured.</p> <p>The Security domain to which the VCS users belong.</p> <p>The VCS Authentication Service uses this environment variable to authenticate VCS users on a remote host.</p> <p>Default: Fully qualified host name of the remote host as defined in the VCS_HOST environment variable or in the .vcshost file.</p>
VCS_DOMAINTYPE	<p>Type of domain: unixpwd, nt, nis, nisplus, or vx.</p> <p>The type of Security domain such as unixpwd, nt, nis, nisplus, ldap, or vx.</p> <p>The VCS Authentication Service uses this environment variable to authenticate VCS users on a remote host.</p> <p>Default: unixpwd</p>
VCS_DIAG	Directory where VCS dumps HAD cores.
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.
VCS_HOME	<p>Root directory for VCS executables.</p> <p>Default: Install Drive:\Program Files\VERITAS\cluster server\</p> <p>Note: You cannot modify this variable.</p>
VCS_HOST	VCS node on which ha commands will be run.

Table 3-2 VCS environment variables (*continued*)

Environment Variable	Definition and Default Value
VCS_GAB_PORT	GAB port to which VCS connects. Default: h
VCS_GAB_TIMEOUT	Timeout in milliseconds for HAD to send heartbeats to GAB. Default: 30000 (denotes 30 seconds) Range: 30000 to 300000 (denotes 30 seconds to 300 seconds) If you set VCS_GAB_TIMEOUT to a value outside the range, the value is automatically reset to 30000 or 300000, depending on the proximity of the value to the lower limit or upper limit of the range. For example, the value is reset to 30000 if you specify 22000 and to 300000 if you specify 400000. Irrespective of the values set, VCS_GAB_TIMEOUT_SECS overrides VCS_GAB_TIMEOUT if both are specified. Note: If the specified timeout is exceeded, GAB kills HAD, and all active service groups on the system are disabled.
VCS_GAB_TIMEOUT_SECS	Timeout in seconds for HAD to send heartbeats to GAB under normal system load conditions. Default: 30 seconds Range: 30 seconds to 300 seconds If you set VCS_GAB_TIMEOUT_SECS to a value outside the range, the value is automatically reset to 30 or 300, depending on the proximity of the value to the lower limit or upper limit of the range. For example, the value is reset to 30 if you specify 22 and to 300 if you specify 400. Irrespective of the values set, VCS_GAB_TIMEOUT_SECS overrides VCS_GAB_TIMEOUT if both are specified. Note: If the specified timeout is exceeded, GAB kills HAD, and all active service groups on the system are disabled.

Table 3-2 VCS environment variables (*continued*)

Environment Variable	Definition and Default Value
VCS_GAB_PEAKLOAD_TIMEOUT_SECS	<p>Timeout in seconds for HAD to send heartbeats to GAB under peak system load conditions.</p> <p>Default: 30 seconds</p> <p>Range: 30 seconds to 300 seconds</p> <p>To set the GAB tunables in adaptive mode, you must set VCS_GAB_PEAKLOAD_TIMEOUT_SECS to a value that exceeds VCS_GAB_TIMEOUT_SECS. If you set VCS_GAB_PEAKLOAD_TIMEOUT_SECS to a value that is lower than VCS_GAB_TIMEOUT_SECS, it is reset to VCS_GAB_TIMEOUT_SECS.</p> <p>Note: If the specified timeout is exceeded, GAB kills HAD, and all active service groups on the system are disabled.</p>
VCS_GAB_RMTIMEOUT	<p>Timeout in milliseconds for HAD to register with GAB.</p> <p>Default: 200000 (denotes 200 seconds)</p> <p>If you set VCS_GAB_RMTIMEOUT to a value less than 200000, the value is automatically reset to 200000.</p> <p>See “About GAB client registration monitoring” on page 461.</p>
VCS_GAB_RMACTION	<p>Controls the GAB behavior when VCS_GAB_RMTIMEOUT exceeds.</p> <p>You can set the value as follows:</p> <ul style="list-style-type: none">■ panic—GAB panics the system■ SYSLOG—GAB logs an appropriate message <p>Default: SYSLOG</p> <p>See “About GAB client registration monitoring” on page 461.</p>
VCS_HAD_RESTART_TIMEOUT	<p>Set this variable to designate the amount of time the hashadow process waits (sleep time) before restarting HAD.</p> <p>Default: 0</p>
VCS_LOG	<p>Root directory for log files and temporary files.</p> <p>You must not set this variable to "" (empty string).</p> <p>Default: Install Drive:\Program Files\VERITAS\cluster server\</p> <p>Note: If this variable is added or modified, you must reboot the system to apply the changes.</p>

Table 3-2 VCS environment variables (*continued*)

Environment Variable	Definition and Default Value
VCS_SERVICE	<p>Name of the configured VCS service.</p> <p>The VCS engine uses this variable to determine the external communication port for VCS. By default, the external communication port for VCS is 14141.</p> <p>The value for this environment variable is defined in the service file at the following location:</p> <p>C:\Windows\System32\drivers\etc\services</p> <p>Default value: vcs</p> <p>If a new port number is not specified, the VCS engine starts with port 14141.</p> <p>To change the default port number, you must create a new entry in the service file at:</p> <p>C:\Windows\System32\drivers\etc\services</p> <p>For example, if you want the external communication port for VCS to be set to 14555, then you must create the following entries in the services file:</p> <p>vcs 14555/tcp</p> <p>vcs 14555/udp</p> <p>Note: The cluster-level attribute OpenExternalCommunicationPort determines whether the port is open or not.</p> <p>See “Cluster attributes” on page 583.</p>
VCS_TEMP_DIR	<p>Directory in which temporary information required by, or generated by, hacf is stored.</p> <p>Default: Install Drive:\Program Files\VERITAS\cluster server\</p> <p>This directory is created in the <code>tmp</code> directory under the following conditions:</p> <ul style="list-style-type: none"> ■ The variable is not set. ■ The variable is set but the directory to which it is set does not exist. ■ The <code>hacf</code> utility cannot find the default location.

Administration - Putting VCS to work

- [Chapter 4. About the VCS user privilege model](#)
- [Chapter 5. Getting started with VCS](#)
- [Chapter 6. Administering the cluster from the command line](#)
- [Chapter 7. Configuring resources and applications in VCS](#)
- [Chapter 8. Modifying the cluster configuration](#)

About the VCS user privilege model

This chapter includes the following topics:

- [About VCS user privileges and roles](#)
- [How administrators assign roles to users](#)
- [User privileges for OS user groups for clusters running in secure mode](#)
- [VCS privileges for users with multiple roles](#)

About VCS user privileges and roles

Cluster operations are enabled or restricted depending on the privileges with which you log on. VCS has three privilege levels: Administrator, Operator, and Guest. VCS provides some predefined user roles; each role has specific privilege levels. For example, the role Guest has the fewest privileges and the role Cluster Administrator has the most privileges.

See [“About administration matrices”](#) on page 511.

VCS privilege levels

[Table 4-1](#) describes the VCS privilege categories.

Table 4-1 VCS privileges

VCS privilege levels	Privilege description
Administrators	Can perform all operations, including configuration

Table 4-1 VCS privileges (*continued*)

VCS privilege levels	Privilege description
Operators	Can perform specific operations on a cluster or a service group.
Guests	Can view specified objects.

User roles in VCS

[Table 4-2](#) lists the predefined VCS user roles, with a summary of their associated privileges.

Table 4-2 User role and privileges

User Role	Privileges
Cluster administrator	<p>Cluster administrators are assigned full privileges. They can make configuration read-write, create and delete groups, set group dependencies, add and delete systems, and add, modify, and delete users. All group and resource operations are allowed. Users with Cluster administrator privileges can also change other users' privileges and passwords.</p> <p>To stop a cluster, cluster administrators require administrative privileges on the local system.</p> <p>Note: Cluster administrators can change their own and other users' passwords only after they change the configuration to read or write mode.</p> <p>Cluster administrators can create and delete resource types.</p>
Cluster operator	<p>Cluster operators can perform all cluster-level, group-level, and resource-level operations, and can modify the user's own password and bring service groups online.</p> <p>Note: Cluster operators can change their own passwords only if configuration is in read or write mode. Cluster administrators can change the configuration to the read or write mode.</p> <p>Users with this role can be assigned group administrator privileges for specific service groups.</p>
Group administrator	<p>Group administrators can perform all service group operations on specific groups, such as bring groups and resources online, take them offline, and create or delete resources. Additionally, users can establish resource dependencies and freeze or unfreeze service groups. Note that group administrators cannot create or delete service groups.</p>

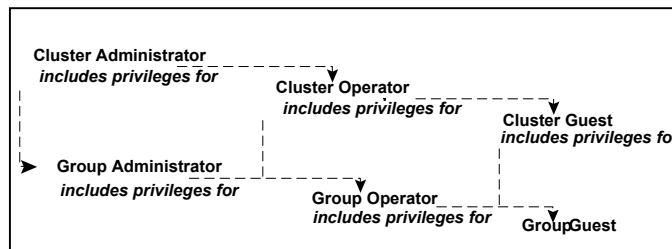
Table 4-2 User role and privileges (*continued*)

User Role	Privileges
Group operator	Group operators can bring service groups and resources online and take them offline. Users can also temporarily freeze or unfreeze service groups.
Cluster guest	Cluster guests have read-only access to the cluster, which means that they can view the configuration, but cannot change it. They can modify their own passwords only if the configuration is in read or write mode. They cannot add or update users. Additionally, users with this privilege can be assigned group administrator or group operator privileges for specific service groups. Note: By default, newly created users are assigned cluster guest permissions.
Group guest	Group guests have read-only access to the service group, which means that they can view the configuration, but cannot change it. The group guest role is available for clusters running in secure mode.

Hierarchy in VCS roles

Figure 4-1 shows the hierarchy in VCS and how the roles overlap with one another.

Figure 4-1 VCS roles



For example, cluster administrator includes privileges for group administrator, which includes privileges for group operator.

User privileges for CLI commands

Users logged with administrative or root privileges are granted privileges that exceed those of cluster administrator, such as the ability to start and stop a cluster.

See “[Cluster attributes](#)” on page 583.

User privileges for cross-cluster operations

A user or user group can perform a cross-cluster online or offline operation only if they have one of the following privileges on the remote cluster:

- Group administrator or group operator privileges for the group
- Cluster administrator or cluster operator privileges

A user or user group can perform a cross-cluster switch operation only if they have one of the following privileges on both the clusters:

- Group administrator or group operator privileges for the group
- Cluster administrator or cluster operator privileges

User privileges for clusters that run in secure mode

In secure mode, VCS assigns guest privileges to all native users.

When you assign privileges for clusters running in secure mode, you must specify fully-qualified user names, in the format `username@domain`.

Note: User names provided in the `domain\username` or `username@domain.com` formats do not work.

In a secure cluster, if the external communication port for VCS is not open, only root users logged on to the host system can run CLI commands.

See [“Cluster attributes”](#) on page 583.

You cannot assign or change passwords for users that use VCS when VCS runs in secure mode.

How administrators assign roles to users

To assign a role to a user, an administrator performs the following tasks:

- Adds a user to the cluster, if the cluster is not running in secure mode.
- Assigns a role to the user.
- Assigns the user a set of objects appropriate for the role. For clusters that run in secure mode, you also can add a role to an operating system user group. See [“User privileges for OS user groups for clusters running in secure mode”](#) on page 70.

For example, an administrator may assign a user the group administrator role for specific service groups. Now, the user has privileges to perform operations on the specific service groups.

You can manage users and their privileges from the command line or from the graphical user interface.

See [“About managing VCS users from the command line”](#) on page 108.

User privileges for OS user groups for clusters running in secure mode

For clusters that run in secure mode, you can assign privileges to native users individually or at an operating system (OS) user group level.

For example, you may decide that all users that are part of the OS administrators group get administrative privileges to the cluster or to a specific service group. Assigning a VCS role to a user group assigns the same VCS privileges to all members of the user group, unless you specifically exclude individual users from those privileges.

When you add a user to an OS user group, the user inherits VCS privileges assigned to the user group.

Assigning VCS privileges to an OS user group involves adding the user group in one (or more) of the following attributes:

- AdministratorGroups—for a cluster or for a service group.
- OperatorGroups—for a cluster or for a service group.

For example, user Tom belongs to an OS user group: OSUserGroup1.

[Table 4-3](#) shows how to assign VCS privileges. FQDN denotes the fully qualified domain name in these examples.

Table 4-3 To assign user privileges

To assign privileges	At an individual level, configure attribute	To the OS user group, configure attribute
Cluster administrator	cluster (Administrators = {tom@FQDN})	cluster (AdministratorGroups = {OSUserGroup1@FQDN})
Cluster operator	cluster (Operators = {tom@FQDN})	cluster (OperatorGroups = {OSUserGroup1@FQDN})
Cluster guest	Cluster (Guests = {tom@FQDN})	Not applicable

Table 4-3 To assign user privileges (*continued*)

To assign privileges	At an individual level, configure attribute	To the OS user group, configure attribute
Group administrator	group <i>group_name</i> (Administrators = {tom@FQDN})	group <i>group_name</i> (AdministratorGroups = {OSUserGroup1@FQDN})
Group operator	group <i>group_name</i> (Operators = {tom@FQDN})	group <i>group_name</i> (OperatorGroups = {OSUserGroup1@FQDN})
Group guest	Cluster (Guests = {tom@FQDN})	Not applicable

VCS privileges for users with multiple roles

[Table 4-4](#) describes how VCS assigns privileges to users with multiple roles. The scenarios describe user Tom who is part of two OS user groups: OSUserGroup1 and OSUserGroup2.

Table 4-4 VCS privileges for users with multiple roles

Situation and rule	Roles assigned in the VCS configuration	Privileges that VCS grants Tom
<p>Situation: Multiple roles at an individual level.</p> <p>Rule: VCS grants highest privileges (or a union of all the privileges) to the user.</p>	<p>Tom: Cluster administrator</p> <p>Tom: Group operator</p>	Cluster administrator.
<p>Situation: Roles at an individual and OS user group level (secure clusters only).</p> <p>Rule: VCS gives precedence to the role granted at the individual level.</p>	<p>Tom: Group operator</p> <p>OSUserGroup1: Cluster administrator</p>	Group operator

Table 4-4 VCS privileges for users with multiple roles (*continued*)

Situation and rule	Roles assigned in the VCS configuration	Privileges that VCS grants Tom
<p>Situation: Different roles for different OS user groups (secure clusters only).</p> <p>Rule: VCS grants the highest privilege (or a union of all privileges of all user groups) to the user.</p>	<p>OSUserGroup1: Cluster administrators</p> <p>OSUserGroup2: Cluster operators</p>	<p>Cluster administrator</p>
<p>Situation: Roles at an individual and OS user group level (secure clusters only).</p> <p>Rule: VCS gives precedence to the role granted at the individual level.</p> <p>You can use this behavior to exclude specific users from inheriting VCS privileges assigned to their OS user groups.</p>	<p>OSUserGroup1: Cluster administrators</p> <p>OSUserGroup2: Cluster operators</p> <p>Tom: Group operator</p>	<p>Group operator</p>

Getting started with VCS

This chapter includes the following topics:

- [Configuring the cluster using the Cluster Configuration Wizard](#)
- [About configuring a cluster from the command line](#)

Configuring the cluster using the Cluster Configuration Wizard

After installing the software, set up the components required to run Cluster Server. The VCS Cluster Configuration Wizard (VCW) sets up the cluster infrastructure, including LLT and GAB, the user account for the VCS Helper service, and provides an option for configuring the VCS Authentication Service in the cluster. The wizard also configures the ClusterService group, which contains resources for notification and global clusters (GCO). You can also use VCW to modify or delete cluster configurations.

Note: After configuring the cluster you must not change the names of the nodes that are part of the cluster. If you wish to change a node name, run VCW to remove the node from the cluster, rename the system, and then run VCW again to add that system to the cluster.

Note the following prerequisites before you proceed:

- The required network adapters, and SCSI controllers are installed and connected to each system.
To prevent lost heartbeats on the private networks, and to prevent VCS from mistakenly declaring a system down, Veritas recommends disabling the Ethernet auto-negotiation options on the private network adapters. Contact the NIC

manufacturer for details on this process. Veritas recommends removing Internet Protocol TCP/IP from private NICs to lower system overhead.

- Verify that the public network adapters on each node use static IP addresses (DHCP is not supported) and name resolution is configured for each node.
- Veritas recommends that you use three network adapters (two NICs exclusively for the VCS private network and one for the public network) per system. You can implement the second private link as a low-priority link over a public interface. Route each private NIC through a separate hub or switch to avoid single points of failure. Veritas recommends that you disable TCP/IP from private NICs to lower system overhead.

Note: If you wish to use Windows NIC teaming, you must select the Static Teaming mode. Only the Static Teaming mode is currently supported.

- Use independent hubs or switches for each VCS communication network (GAB and LLT). You can use cross-over Ethernet cables for two-node clusters. GAB supports hub-based or switch network paths, or two-system clusters with direct network links.
- Verify the DNS settings for all systems on which the application is installed and ensure that the public adapter is the first adapter in the Connections list. When enabling DNS name resolution, make sure that you use the public network adapters, and not those configured for the VCS private network.
- The logged on user must have local Administrator privileges on the system where you run the wizard. The user account must be a domain user account.
- The logged on user must have administrative access to all systems selected for cluster operations. Add the domain user account to the local Administrator group of each system.
- If you plan to create a new user account for the VCS Helper service, the logged on user must have Domain Administrator privileges or must belong to the Domain Account Operators group.
- When configuring a user account for the Veritas VCS Helper service, make sure that the user account is a domain user. The Veritas High Availability Engine (HAD), which runs in the context of the local system built-in account, uses the Veritas VCS Helper service user context to access the network. This account does not require Domain Administrator privileges.
- Make sure the VCS Helper service domain user account has "Add workstations to domain" privilege enabled in the Active Directory.

- Verify that each system can access the storage devices and each system recognizes the attached shared disk.
Use Windows Disk Management on each system to verify that the attached shared LUNs (virtual disks) are visible.
- If you plan to set up a disaster recovery (DR) environment, you must configure the wide-area connector process for global clusters.
- If you are setting up a Replicated Data Cluster configuration, add only the systems in the primary zone (zone 0) to the cluster, at this time.

To configure a VCS cluster using the wizard

- 1 Start the VCS Cluster Configuration Wizard from **Start > All Programs > Veritas > Veritas Cluster Server > Configuration Tools > Cluster Configuration Wizard** or, on Windows Server 2012 operating systems, from the **Apps** menu in the **Start** screen.
- 2 Read the information on the Welcome panel and click **Next**.
- 3 On the Configuration Options panel, click **Cluster Operations** and click **Next**.
- 4 On the Domain Selection panel, select or type the name of the domain in which the cluster resides and select the discovery options.

To discover information about all systems and users in the domain, do the following:

- Clear **Specify systems and users manually**.
- Click **Next**.
Proceed to step 8.

To specify systems and user names manually (recommended for large domains), do the following:

- Select **Specify systems and users manually**.
Additionally, you may instruct the wizard to retrieve a list of systems and users in the domain by selecting appropriate check boxes.
- Click **Next**.
If you chose to retrieve the list of systems, proceed to step 6. Otherwise, proceed to the next step.

- 5 On the System Selection panel, type the name of each system to be added, click **Add**, and then click **Next**.

Do not specify systems that are part of another cluster.

Proceed to step 8.

- 6 On the System Selection panel, specify the systems for the cluster and then click **Next**.

Do not select systems that are part of another cluster.

Enter the name of the system and click **Add** to add the system to the Selected Systems list, or click to select the system in the Domain Systems list and then click the > (right-arrow) button.

- 7 The System Report panel displays the validation status, whether Accepted or Rejected, of all the systems you specified earlier. Review the status and then click **Next**.

Select the system to see the validation details. If you wish to include a rejected system, rectify the error based on the reason for rejection and then run the wizard again.

A system can be rejected for any of the following reasons:

- System is not pingable.
- WMI access is disabled on the system.
- Wizard is unable to retrieve the system architecture or operating system.
- Product is either not installed or there is a version mismatch.

- 8 On the Cluster Configuration Options panel, click **Create New Cluster** and then click **Next**.

- 9 On the Cluster Details panel, specify the details for the cluster and then click **Next**.

Veritas Cluster Server Configuration Wizard

Cluster Details
Enter necessary details to create the new cluster

Domain Selection

Cluster Details

Cluster Selection

Validate Systems

Edit Options

NIC Selection

Service Account

Security

Summary

Finish

Specify the cluster name and cluster ID. If you chose to specify the systems manually, VCSW does not validate the cluster ID.

Cluster Name:

Cluster ID:

Operating System:

Select the systems to create the cluster.

☒ **Select all systems**

Available Systems

- ☒ ROGER
- ☒ SCOOPYDU

Total number of systems selected to create the cluster : 2

Click 'Next' to continue.

VERITAS

Specify the cluster details as follows:

- | | |
|------------------|--|
| Cluster Name | Type a name for the new cluster. Veritas recommends a maximum length of 32 characters for the cluster name. |
| Cluster ID | Select a cluster ID from the suggested cluster IDs in the drop-down list, or type a unique ID for the cluster. The cluster ID can be any number from 0 to 65535.

Note: If you chose to specify systems and users manually in step 4 or if you share a private network between more than one domain, make sure that the cluster ID is unique. |
| Operating System | From the drop-down list, select the operating system.

All the systems in the cluster must have the same operating system and architecture. |

Available Systems Select the systems that you wish to configure in the cluster.

Check the **Select all systems** check box to select all the systems simultaneously.

The wizard discovers the NICs on the selected systems. For single-node clusters with the required number of NICs, the wizard prompts you to configure a private link heartbeat. In the dialog box, click **Yes** to configure a private link heartbeat.

- 10** The wizard validates the selected systems for cluster membership. After the systems are validated, click **Next**.

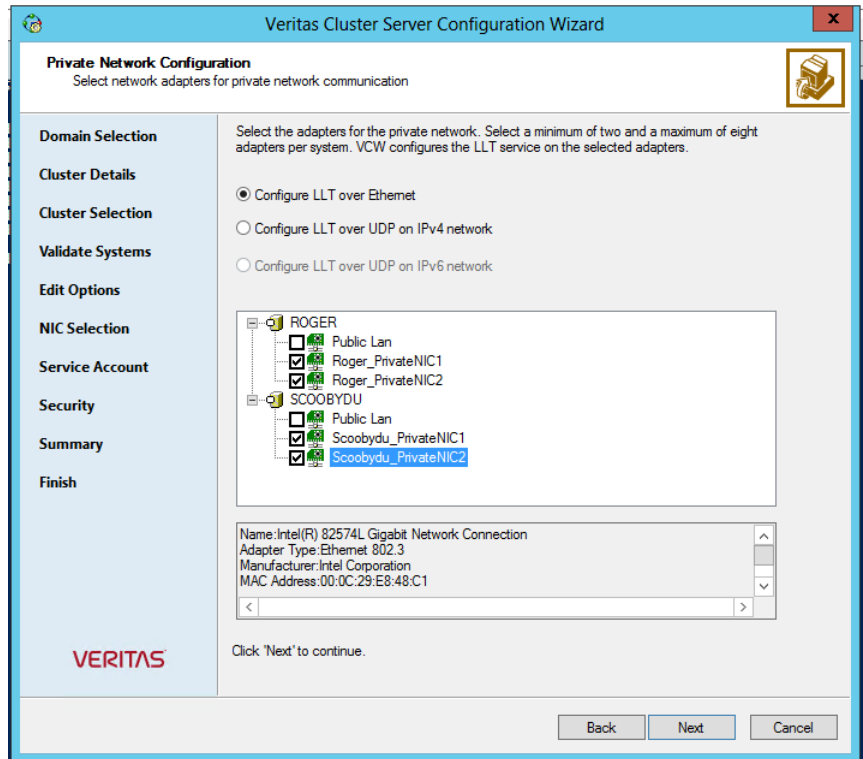
If a system is not validated, review the message associated with the failure and restart the wizard after rectifying the problem.

If you chose to configure a private link heartbeat in step 9, proceed to the next step. Otherwise, proceed to step 12.

- 11** On the Private Network Configuration panel, configure the VCS private network and then click **Next**. You can configure the VCS private network either over the ethernet or over the User Datagram Protocol (UDP) layer using IPv4 or IPv6 network.

Do one of the following:

- To configure the VCS private network over ethernet, complete the following steps:



- Select **Configure LLT over Ethernet**.
- Select the check boxes next to the two NICs to be assigned to the private network. You can assign a maximum of eight network links. Veritas recommends reserving two NICs exclusively for the private network. However, you could lower the priority of one of the NICs and use the low-priority NIC for both public and as well as private communication.
- If there are only two NICs on a selected system, Veritas recommends that you lower the priority of at least one NIC that will be used for private as well as public network communication. To lower the priority of a NIC, right-click the NIC and select **Low Priority** from the pop-up menu.
- If your configuration contains teamed NICs, the wizard groups them as "NIC Group #N" where "N" is a number assigned to the teamed NIC. A teamed NIC is a logical NIC, formed by grouping several physical NICs together. All NICs in a team have an identical MAC address. Veritas recommends that you do not select teamed NICs for the private network.

The wizard configures the LLT service (over ethernet) on the selected network adapters.

- To configure the VCS private network over the User Datagram Protocol (UDP) layer, complete the following steps:
- Select **Configure LLT over UDP on IPv4 network** or **Configure LLT over UDP on IPv6 network** depending on the IP protocol that you wish to use. The IPv6 option is disabled if the network does not support IPv6.
- Select the check boxes next to the NICs to be assigned to the private network. You can assign a maximum of eight network links. Veritas recommends reserving two NICs exclusively for the VCS private network.
- For each selected NIC, verify the displayed IP address. If a selected NIC has multiple IP addresses assigned, double-click the field and choose the desired IP address from the drop-down list. In case of IPv4, each IP address can be in a different subnet.

The IP address is used for the VCS private communication over the specified UDP port.

- Specify a unique UDP port for each of the link. Click **Edit Ports** if you wish to edit the UDP ports for the links. You can use ports in the range 49152 to 65535. The default ports numbers are 50000 and 50001 respectively. Click **OK**.

For each selected NIC, double-click the respective field in the Link column and choose a link from the drop-down list. Specify a different link (Link1 or Link2) for each NIC. Each link is associated with a UDP port that you specified earlier.

The wizard configures the LLT service (over UDP) on the selected network adapters. The specified UDP ports are used for the private network communication.

- 12** On the VCS Helper Service User Account panel, specify the name of a domain user for the VCS Helper service.

The Veritas High Availability Engine (HAD), which runs in the context of the local system built-in account, uses the Veritas VCS Helper service user context to access the network. This account does not require Domain Administrator privileges.

Specify the domain user details as follows:

- To specify an existing user, do one of the following:
 - Click **Existing user** and select a user name from the drop-down list.
 - If you chose not to retrieve the list of users in step 4, type the user name in the Specify User field and then click **Next**.

- To specify a new user, click **New user** and type a valid user name in the Create New User field and then click **Next**.
Do not append the domain name to the user name; do not type the user name as Domain\user or user@domain.
- In the Password dialog box, type the password for the specified user and click **OK**, and then click **Next**.

13 On the Configure Security Service Option panel, specify security options for the cluster communications and then click **Next**.

Do one of the following:

- To use VCS cluster user privileges, click **Use VCS User Privileges** and then type a user name and password.
The wizard configures this user as a VCS Cluster Administrator. In this mode, communication between cluster nodes and clients, including Cluster Manager (Java Console), occurs using the encrypted VCS cluster administrator credentials. The wizard uses the VCSEncrypt utility to encrypt the user password.
The default user name for the VCS administrator is *admin* and the password is *password*. Both are case-sensitive. You can accept the default user name and password for the VCS administrator account or type a new name and password.
Veritas recommends that you specify a new user name and password.
- To use the single sign-on feature, click **Use Single Sign-on**.
In this mode, the VCS Authentication Service is used to secure communication between cluster nodes and clients by using digital certificates for authentication and SSL to encrypt communication over the public network. VCS uses SSL encryption and platform-based authentication. The Veritas High Availability Engine (HAD) and Veritas Command Server run in secure mode.
The wizard configures all the cluster nodes as root brokers (RB) and authentication brokers (AB). Authentication brokers serve as intermediate registration and certification authorities. Authentication brokers have certificates signed by the root. These brokers can authenticate clients such as users and services. The wizard creates a copy of the certificates on all the cluster nodes.

- 14** Review the summary information on the Summary panel, and click **Configure**.

The wizard configures the VCS private network. If the selected systems have LLT or GAB configuration files, the wizard displays an informational dialog box before overwriting the files. In the dialog box, click **OK** to overwrite the files. Otherwise, click **Cancel**, exit the wizard, move the existing files to a different location, and rerun the wizard.

The wizard starts running commands to configure VCS services. If an operation fails, click **View configuration log file** to see the log.

- 15** On the Completing Cluster Configuration panel, click **Next** to configure the ClusterService group; this group is required to set up components for notification and for global clusters.

To configure the ClusterService group later, click **Finish**.

At this stage, the wizard has collected the information required to set up the cluster configuration. After the wizard completes its operations, with or without the ClusterService group components, the cluster is ready to host application service groups. The wizard also starts the VCS engine (HAD) and the Veritas Command Server at this stage.

- 16** On the Cluster Service Components panel, select the components to be configured in the ClusterService group and then click **Next**.

Do the following:

- Check the **Notifier Option** check box to configure notification of important events to designated recipients.
See [“Configuring notification”](#) on page 82.
- Check the **GCO Option** check box to configure the wide-area connector (WAC) process for global clusters. The WAC process is required for inter-cluster communication.
Configure the GCO Option using this wizard only if you are configuring a Disaster Recovery (DR) environment and are not using the Disaster Recovery wizard.
You can configure the GCO Option using the DR wizard. The Disaster Recovery chapters in the application solutions guides discuss how to use the Disaster Recovery wizard to configure the GCO option.
See [“Configuring Wide-Area Connector process for global clusters”](#) on page 85.

Configuring notification

This section describes steps to configure notification.

To configure notification

- 1 On the Notifier Options panel, specify the mode of notification to be configured and then click **Next**.

You can configure VCS to generate SNMP (V2) traps on a designated server and send emails to designated recipients in response to certain events.

- 2 If you chose to configure SNMP, specify information about the SNMP console and then click **Next**.

The screenshot shows the 'Notifier SNMP Configuration' window of the Veritas Cluster Server Configuration Wizard. The window has a blue title bar and a sidebar on the left with navigation links: Domain Selection, Create Cluster, Select Components, Configure, Summary, and Finish. The main area is titled 'Notifier SNMP Configuration' with the subtitle 'Specify information about SNMP console.' Below this, it says 'Enter the name or the IP address of the SNMP console and then select the desired severity level.' There is a table with two columns: 'SNMP Console' and 'Severity Information'. The 'SNMP Console' column has a text input field with the placeholder 'Click here to change the text..'. The 'Severity Information' column has a dropdown menu. Below the table, there are instructions: 'Click on '+' button to add more consoles.' and 'Click '-' to remove a console.' with corresponding '+' and '-' buttons. There is also an 'SNMP Trap Port' field with the value '162'. A note states: 'Note: SNMP console must be MIB 2.0 compliant.' and a prompt says 'Click 'Next' to continue.' At the bottom right, there are 'Back', 'Next', and 'Cancel' buttons. The 'Next' button is highlighted with a mouse cursor.

Do the following:

- Click a field in the **SNMP Console** column and type the name or IP address of the console.
The specified SNMP console must be MIB 2.0 compliant.
- Click the corresponding field in the **Severity** column and select a severity level for the console.
- Click the + icon to add a field; click the - icon to remove a field.

- Enter an SNMP trap port. The default value is 162.
- 3 If you chose to configure SMTP, specify information about SMTP recipients and then click **Next**.

Veritas Cluster Server Configuration Wizard

Notifier SMTP Configuration
Specify information about SMTP recipients.

Domain Selection

Create Cluster

Select Components

Configure

Summary

Finish

SMTP Server Name / IP

Enter SMTP recipients and select a severity level for each recipient.

Recipients	Severity
Click here to change the text..	Information

Click '+' to add a recipient.
Click '-' to remove a recipient.

Click 'Next' to continue.

Back Next Cancel

Do the following:

- Type the name of the SMTP server.
 - Click a field in the **Recipients** column and enter a recipient for notification. Enter recipients as admin@example.com.
 - Click the corresponding field in the **Severity** column and select a severity level for the recipient.
VCS sends messages of an equal or higher severity to the recipient.
 - Click the + icon to add fields; click the - icon to remove a field.
- 4 On the Notifier Network Card Selection panel, specify the network information and then click **Next**.

Do the following:

- If the cluster has a ClusterService group configured, you can use the NIC resource configured in that service group or configure a new NIC resource for notification.
 - If you choose to configure a new NIC resource, select a network adapter for each node in the cluster.
The wizard lists the public network adapters along with the adapters that were assigned a low priority.
- 5 Review the summary information and choose whether you want to bring the notification resources online when VCS starts and click **Configure**.
 - 6 Click **Finish** to exit the wizard.

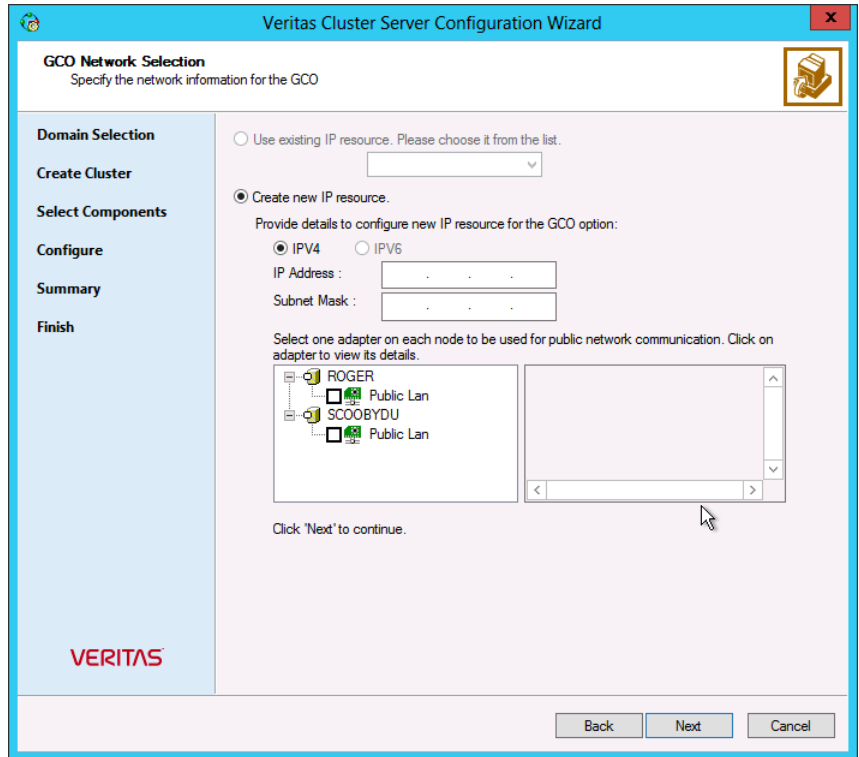
Configuring Wide-Area Connector process for global clusters

Configure the Wide-Area Connector process only if you are configuring a disaster recovery environment. The GCO option configures the wide-area connector (WAC) process for global clusters. The WAC process is required for inter-cluster communication. Configure the GCO Option using this wizard only if you are configuring a Disaster Recovery (DR) environment and are not using the Disaster Recovery wizard.

You can configure the GCO Option using the DR wizard. The Disaster Recovery chapters in the application solutions guides discuss how to use the Disaster Recovery wizard to configure the GCO option.

To configure the wide-area connector process for global clusters

- 1 On the GCO Network Selection panel, specify the network information and then click **Next**.



If the cluster has a ClusterService group configured, you can use the IP address configured in the service group or configure a new IP address.

Do the following:

- To specify an existing IP address, select **Use existing IP resource** and then select the IP address from the drop-down list.
- To use a new IP address, do the following:
 - In case of IPv4, select **IPv4** and then enter the IP address and associated subnet mask. Make sure that the specified IP address has a DNS entry.
 - In case of IPv6, select **IPv6** and select the IPv6 network from the drop-down list.
 The wizard uses the network prefix and automatically generates a unique IPv6 address that is valid on the network.
 The IPv6 option is disabled if the network does not support IPv6.
- Select a network adapter for each node in the cluster.

The wizard lists the public network adapters along with the adapters that were assigned a low priority.

- 2 Review the summary information and choose whether you want to bring the WAC resources online when VCS starts and then click **Configure**.
- 3 Click **Finish** to exit the wizard.

About configuring a cluster from the command line

VCS provides a silent configuration utility, `VCWsilent.exe`, which lets you configure a new cluster and delete an existing cluster. You can use `VCWsilent.exe` to perform these tasks only on one cluster at a time.

About preparing for a silent configuration

To configure or delete a cluster, the silent configuration utility requires an XML configuration file that contains information about the cluster. No such file is required when re-configuring a cluster for single sign-on authentication.

About configuring a non-secure cluster

The XML file must have the following format for configuring a non-secure cluster:

```
<Operation Type="New">
  <Domain Name="domain_name">
    <SystemList>
      <System Name="sys_name1"/>
      <System Name="sys_name2"/>
      ....
      ....
    </SystemList>
    <Cluster Name="clus_name" ID="clus_ID" SingleNode="SingleNodeValue">
      <Node Name="sys_name1">
        <LLTLink Name="adp_name_1" MAC="MAC_address_1"
          LowPri="pri"/>
        <LLTLink Name="adp_name_2"
          MAC="MAC_address_2" LowPri="pri"/>
      </Node>
      <Node Name="sys_name2">
        <LLTLink Name="adp_name_1"
          MAC="MAC_address_1" LowPri="pri"/>

```

```

        <LLTLink Name="adp_name_2"
MAC="MAC_address_2" LowPri="pri"/>
    </Node>

    ....

    ....
    <Security Type="Non-Secured">
        <Admin User="admin_user_name" Password="password"/>
    </Security>
    <HadHelperUser Name="HAD_user_name" Password="password"/>
</Cluster>
</Domain>
</Operation>

```

About configuring a secure cluster

The XML file must have the following format for configuring a secure cluster:

```

<Operation Type="New">
    <Domain Name="domain_name">
        <SystemList>
            <System Name="sys_name_1"/>
            <System Name="sys_name_2"/>
            ....
            ....
        </SystemList>
        <Cluster Name="clus_name" ID="clus_ID" SingleNode="SingleNodeValue">
            <Node Name="node_name_1">
                <LLTLink Name="adp_name_1"
MAC="MAC_address_1" LowPri="pri"/>
                <LLTLink Name="adp_name_2"
MAC="MAC_address_2" LowPri="pri"/>
            </Node>
            <Node Name="node_name_2">
                <LLTLink Name="adp_name_1"
MAC="MAC_address_1" LowPri="pri"/>
                <LLTLink Name="adp_name_2"
MAC="MAC_address_2" LowPri="pri"/>
            </Node>
            ....
            ....
            <Security Type="Secured">
                <VxSSRoot Name="root_name"/>
            </Security>
            <HadHelperUser Name="HAD_user_name" Password="password"/>
        </Cluster>
    </Domain>
</Operation>

```



```

        </Cluster>
    </Domain>
</Operation>

```

About deleting a non-secure cluster

The XML file must have the following format for deleting a non-secure cluster:

```

<Operation Type="Delete">
    <Domain Name="domain_name">
        <SystemList>
            <System Name="sys_name1"/>
            <System Name="sys_name2"/>
            ....
            ....
        </SystemList>
        <Cluster Name="clus_name" ID="clus_ID"
            ConnecttoCluster="ConnecttoClustervalue"
            IgnoreOfflineGroups="IgnoreOfflineGroupsvalue">
            <Security Type="Non-Secured">
                <Admin User="admin_user_name" Password="password"/>
            </Security>
            <HadHelperUser Remove="Removevalue"
                Name="HAD_user_name" Password="password"/>
        </Cluster>
    </Domain>
</Operation>

```

About deleting a secure cluster

The XML file must be of the following format for deleting a secure cluster:

```

<Operation Type="Delete">
    <Domain Name="domain_name">
        <SystemList>
            <System Name="sys_name_1"/>
            <System Name="sys_name_2"/>
            ....
            ....
        </SystemList>
        <Cluster Name="clus_name" ID="clus_ID"
            ConnecttoCluster="ConnecttoClustervalue"
            IgnoreOfflineGroups="IgnoreOfflineGroupsvalue">
            <Security Type="Secured">

```

```

        <VxSSRoot Name="root_name" />
    </Security>
    <HadHelperUser Remove="Removevalue" Name="HAD_user_name"
        Password="password" />
    </Cluster>
    </Domain>
</Operation>

```

Copy the relevant format to any text editor and save it with a .xml extension. Replace the variables, shown in *italics*, with appropriate values. Review the information about variables and their possible values.

See [“About element attributes values”](#) on page 90.

A sample XML file is included for your reference.

See [“About sample XML configuration”](#) on page 93.

About element attributes values

[Table 5-1](#) describes the variables that are used in the XML format and their possible values:

* "n" is the sequence number for the systems, nodes, adapters, and MAC addresses.

Table 5-1 VCWsilent - variables and values

Variables	Description
<i>domain_name</i>	Replace this variable with the fully qualified name of a domain in which the systems reside.
<i>sys_name_<n*></i>	Replace this with name of the system in the domain for which relevant information will be discovered. Note: For each system, you must have a System child element under the SystemList element.
<i>clus_name</i>	Replace this with the name of the cluster to be created.
<i>clus_ID</i>	Replace this with the cluster ID. Make sure you specify a unique cluster ID between 0 and 65535.
<i>SingleNodeValue</i>	Replace this "1" or "0." The value "1" indicates that it is a single node cluster. The value "0" indicates that it is a multi-node cluster.

Table 5-1 VCWsilent - variables and values (*continued*)

Variables	Description
<i>node_name_<n*></i>	<p>Replace this with the name of the system that will be part of the cluster. Make sure that you provide system names from the list of systems that are specified under the SystemList element.</p> <p>For example, if you specified SysA and SysB in the SystemList element, you can specify one or both the systems for the node names. However, you should not specify another system, say SysC, which was not specified in the SystemList element.</p> <p>Note: For each node, you must have a Node child element along with the LLTLink subchild element under the Cluster element.</p>
<i>adp_name_<n*></i>	<p>Replace this with the name of the adapter where the LLT link will be configured.</p> <p>Note: For each node, you must specify a minimum of two adapters. Each adapter must be specified as an attribute of the LLTLink element.</p>
<i>MAC_address_<n*></i>	<p>Replace this with the MAC address of the adapter.</p>
<i>Pri</i>	<p>Replace this with either "1" or "0." Value "1" indicates that the adapter is assigned a low priority. Value "0" indicates otherwise. You can assign a low priority to an adapter to use it for both private and public network communication.</p>
<i>admin_user_name</i>	<p>Replace this with a user name for the cluster administrator. You can use this user name to log on to a cluster that uses Cluster Manager.</p> <p>Note: This user name is applicable only for a non-secure cluster.</p>
<i>root_name</i>	<p>Replace this with the host name of one of the systems selected for the cluster configuration. It should be one of the systems specified for the SystemList element.</p> <p>Note: This system name is applicable only for a secure cluster.</p>

Table 5-1 VCWsilent - variables and values (*continued*)

Variables	Description
<i>HAD_user_name</i>	Replace this with a domain user name in whose context the VCS Helper service will run. The VCS High Availability Daemon, which runs in the context of the local system built-in account, uses the VCS Helper service user context to access the network.
<i>password</i>	Replace this with an encrypted password. See “About encrypting passwords” on page 92.
<i>ConnecttoClustervalue</i>	Replace this with "Yes" if you want to connect to the cluster before you delete it. If the connection fails, the cluster deletion does not proceed. The default value is "No." This default value indicates that the cluster is deleted without connecting to it.
<i>IgnoreOfflineGroupsvalue</i>	Replace this with "Yes" if you wish to delete the cluster along with the service groups that are configured in the cluster. The default value is "No". This means that the cluster deletion does not proceed if there are service groups in the cluster.
<i>Removevalue</i>	Replace this with "Yes" if you want to delete the VCS Helper Service account along with the cluster. The default value is "No." This default value indicates that the VCS Helper account will not be deleted.

About encrypting passwords

Before you specify passwords in the XML configuration file, you must encrypt them by using the `vcseencrypt` utility.

Note: If UAC is enabled, run the program or commands in the “Run as administrator” mode even if the logged-on user belongs to the local administrators group. Alternatively, log on as an Administrator (default administrator account) to perform the tasks.

You must encrypt all the passwords to be specified in the XML file.

See [“Encrypting agent passwords”](#) on page 99.

Specify only the encrypted passwords in the XML file.

About sample XML configuration

Sample XML configuration files are provided for reference.

For two-node secure cluster configuration:

Use this configuration file to create a secure cluster with systems SYSTEM1 and SYSTEM2.

```
<Operation Type="New">
  <Domain Name="DOMAIN.com">
    <SystemList>
      <System Name="SYSTEM1"/>
      <System Name="SYSTEM2"/>
    </SystemList>
    <Cluster Name="MYCLUSTER" ID="0">
      <Node Name="SYSTEM1">
        <LLTLink Name="Adapter0" MAC="00:03:47:08:91:56"
        LowPri="0"/>
        <LLTLink Name="Adapter1" MAC="00:03:47:08:91:C6"
        LowPri="0"/>
      </Node>
      <Node Name="SYSTEM2">
        <LLTLink Name="Adapter0" MAC="00:03:47:08:91:CC"
        LowPri="0"/>
        <LLTLink Name="Adapter1" MAC="00:03:47:08:94:4E"
        LowPri="0"/>
      </Node>
      <Security Type="Secured">
        <VxSSRoot Name="SYSTEM1"/>
      </Security>
      <HadHelperUser Name="Administrator" Password="hvnTkvK"/>
    </Cluster>
  </Domain>
</Operation>
```

For two-node secure cluster deletion:

Use this configuration file to delete a secure cluster with systems SYSTEM1 and SYSTEM2.

```
<Operation Type="Delete">
  <Domain Name="DOMAIN.com">
    <SystemList>
      <System Name="SYSTEM1"/>
      <System Name="SYSTEM2"/>
    </SystemList>
  </Domain>
</Operation>
```

```

</SystemList>
<Cluster Name="MYCLUSTER" ID="0" ConnecttoCluster="No"
  IgnoreOfflineGroups="Yes">
  <Security Type="Secured">
    <VxSSRoot Name="SYSTEM1"/>
  </Security>
  <HadHelperUser Remove="No" Name="
    Administrator" Password="hvnTkVK"/>
  </Cluster>
</Domain>
</Operation>

```

Running the silent configuration utility

Review the prerequisites before you run the silent configuration utility, `VCWsilent.exe`.

- If UAC is enabled, run the program or commands in the “Run as administrator” mode even if the logged-on user belongs to the local administrators group. Alternatively, log on as an Administrator (default administrator account) to perform the tasks.
- To configure or delete a cluster, run the utility from any system in the domain, irrespective of whether the system is part of the cluster.

To run the silent configuration utility

- 1 At the command line, type one of the following commands:

- To configure a new cluster or delete an existing cluster, use:

```
command_prompt> VCWsilent XML_file_name_including_path
```

- To view the progress when configuring or deleting a cluster, use the `-v` option:

```
command_prompt> VCWsilent XML_file_name_including_path -v
```

- 2 If the cluster is successfully configured or deleted, the following message appears:

```
Silent configuration was successful.
```

If the silent configuration fails, an error message appears. Review the message associated with the failure and rerun the utility after you rectify the problem.

To reconfigure or upgrade a cluster using the silent configuration utility

- 1** Back up your cluster configuration by creating a copy of the `main.cf` file.
- 2** Use the `VCWsilent` utility to perform the following operations sequentially:
 - Delete the cluster.
 - Create a new cluster.
- 3** Stop the cluster using the following command:

```
hastop -all -force
```

- 4** Manually restore your application configurations by copying only the relevant entries from the backed-up `main.cf` to newly created `main.cf`.
- 5** Start the cluster using the following command:

```
hastart
```

Administering the cluster from the command line

This chapter includes the following topics:

- [About administering VCS from the command line](#)
- [Starting VCS](#)
- [Stopping the VCS engine and related processes](#)
- [About managing VCS configuration files](#)
- [About managing VCS users from the command line](#)
- [About querying VCS](#)
- [About administering service groups](#)
- [Administering agents](#)
- [About administering resources](#)
- [About administering resource types](#)
- [Administering systems](#)
- [About administering clusters](#)
- [Using the -wait option in scripts that use VCS commands](#)
- [About administering simulated clusters from the command line](#)

About administering VCS from the command line

Review the details on commonly used commands to administer VCS. For more information about specific commands or their options, see their usage information or the man pages associated with the commands.

You can enter most commands from any system in the cluster when VCS is running. The command to start VCS is typically initiated at system startup.

Note: If UAC is enabled, run the program or commands in the “Run as administrator” mode even if the logged-on user belongs to the local administrators group. Alternatively, log on as an Administrator (default administrator account) to perform the tasks.

Symbols used in the VCS command syntax

Table 6-1 specifies the symbols used in the VCS commands. Do not use these symbols when you run the commands.

Table 6-1 Symbols used in the VCS commands

Symbols	Usage	Example
[]	Used for command options or arguments that are optional.	hasys -freeze [-persistent] [-evacuate] <i>system</i>
	Used to specify that only one of the command options or arguments separated with can be used at a time.	hagetcf [-s -silent]
...	Used to specify that the argument can have several values.	hagrp -modify group <i>attribute</i> value ... [-sys <i>system</i>]
{ }	Used to specify that the command options or arguments enclosed within these braces must be kept together.	haatr -display {cluster group system heartbeat <restype>} haclus -modify <i>attribute</i> { <i>key value</i> }
<>	Used in the command help or usage output to specify that these variables must be replaced with the actual values.	haclus -help VCS INFO V-16-1-10601 Usage: haclus -add <cluster> <ip> haclus -delete <cluster>

See [“About administering VCS from the command line”](#) on page 97.

How VCS identifies the local system

VCS uses the system's node name. To view the system's node name from the command line, type:

```
C:\> hostname
```

To view the system's node name from the desktop

- 1 Right-click My Computer to display the pop-up menu.
- 2 Click **Properties**. The name of the system is listed in the Computer Name tab.

About specifying values preceded by a dash (-)

When you specify values in a command-line syntax, you must prefix values that begin 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 the High Availability Daemon (HAD) and does not appear in the configuration file.)

About the -modify option

Most configuration changes are made by using the `-modify` options of the commands `haclus`, `hagrp`, `hares`, `hasys`, and `hatype`. Specifically, the `-modify` option of these commands changes the attribute values that are 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 [“Setting the configuration to read or write”](#) on page 107.

Encrypting VCS passwords

Use the `vcsencrypt` utility to encrypt passwords when you edit the VCS configuration file `main.cf` to add VCS users.

Note: Do not use the `vcsencrypt` utility when you enter passwords from a configuration wizard or from the Java console.

To encrypt a password

- 1 Run the utility from the command line.

```
# vcsencrypt -vcs
```

- 2 The utility prompts you to enter the password twice. Enter the password and press Return.

```
Enter Password:
```

```
Enter Again:
```

- 3 The utility encrypts the password and displays the encrypted password. Use this password to edit the VCS configuration file `main.cf`.

Encrypting agent passwords

To configure agents that require user passwords, you may need to edit the VCS configuration file, `main.cf`. To encrypt any such user passwords, use the `vcsencrypt` utility.

Note: Do not use the `vcsencrypt` utility when you enter passwords from a configuration wizard or from the Java console.

Encrypting agent passwords without using security keys

You may use the native VCS algorithm to encrypt agent passwords as described in the following procedure.

Note: Veritas recommends that you use security keys to encrypt agent passwords, as described in "[Encrypting agent passwords by using security keys](#)," which is a more secure method.

To encrypt an agent password without using a security key

- 1 Run the utility from the command line.

```
vcsencrypt -agent
```

- 2 The utility prompts you to enter the same password twice.

```
Enter New Password:  
Enter Again:
```

After you provide the same password twice, the utility encrypts the password and displays the encrypted password.

- 3 Use this password to edit the VCS configuration file, `main.cf`.

Encrypting agent passwords by using security keys

Encrypting the agent password using security keys involves the following tasks:

- See [“Generating a security key”](#) on page 100.
- See [“Encrypting the agent password”](#) on page 101.
- (Optional) See [“Changing the security key”](#) on page 101.

Generating a security key

Use the `vcsencrypt` utility to generate a security key.

Note: You must be a root user or must have administrative privileges to generate the security keys.

To generate a security key, perform the following steps on any cluster node:

- 1 Make the VCS configuration writable.

```
haconf -makerw
```

- 2 Run the `vcsencrypt` utility.

```
vcsencrypt -gensecinfo
```

- 3 When prompted, enter any pass phrase, which has at least eight characters.

The utility generates the security key and displays the following message:

```
Generating SecInfo...please wait...
SecInfo generated successfully.
Trying to update its value in config file...
SecInfo updated successfully.
```

- 4 Save the VCS configuration and make it read only.

```
haconf -dump -makero
```

Note: This step is crucial; if you do not perform this step, the VCS configuration cannot use the newly generated encryption key.

Encrypting the agent password

Perform the following procedure on any cluster node.

To encrypt the agent password

- 1 Make the VCS configuration writable.

```
haconf -makerw
```

- 2 Encrypt the agent password.

```
vcsencrypt -agent -secinfo
```

- 3 When prompted, type the existing password and press Enter.

The utility encrypts the entered password and displays the encrypted password; for example:

```
0b:c3:0f:7f:42:52:cl:21:eb:7f:0d:59:28:56:e2:32:ab:b7:f7:79:0a:4a:5a:34:be:cl:79:ca:27:cl:bc:58
```

- 4 Modify the required agent attribute to assign the encrypted password.

```
hares -modify resourceName passwordAttributeName AESPassword
```

Changing the security key

If you change the security key, you must use the new security key to re-encrypt all the passwords that were encrypted earlier with the previous security key. Otherwise, the agent fails to decrypt the encrypted password, and may consequently fail to perform resource monitoring.

To change security key

- 1 Save the VCS configuration and make it writeable.

```
haconf -makerw
```

- 2 Run the following command:

```
vcsencrypt -gensecinfo -force
```

- 3 Save the VCS configuration and make it read only.

```
haconf -dump -makero
```

Starting VCS

When VCS starts, 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.

If the configuration on all nodes is invalid, the VCS engine waits for manual intervention, or for VCS to be started on a system that has a valid configuration.

See [“Remote cluster states”](#) on page 519.

See [“System states”](#) on page 521.

To start VCS

- ◆ Run the following command:

```
hastart
```

To start VCS when all systems are in the ADMIN_WAIT state

- ◆ Run 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
```

To start 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.

```
hastart -onenode
```

To start VCS as a time-sharing process

- ◆ Run the following command:

```
hastart -ts
```

To start CommandServer

- ◆ Run the following command:

```
net start cmdserver
```

Stopping the VCS engine and related processes

The `hastop` command stops the High Availability Daemon (HAD) and related processes. You can customize the behavior of the `hastop` command by configuring the `EngineShutdown` attribute for the cluster.

See [“About controlling the hastop behavior by using the EngineShutdown attribute”](#) on page 104.

The `hastop` command includes the following options:

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

[Table 6-2](#) shows the options for the `hastop` command.

Table 6-2 Options for the `hastop` command

Option	Description
-all	Stops HAD on all systems in the cluster and takes all service groups offline.
-help	Displays the command usage.
-local	Stops HAD on the system on which you typed the command
-force	Allows HAD to be stopped without taking service groups offline on the system. The value of the <code>EngineShutdown</code> attribute does not influence the behavior of the <code>-force</code> option.
-evacuate	When combined with <code>-local</code> or <code>-sys</code> , migrates the system's active service groups to another system in the cluster, before the system is stopped.

Table 6-2 Options for the `hastop` command (*continued*)

Option	Description
<code>-noautodisable</code>	Ensures the service groups that can run on the node where the <code>hastop</code> command was issued are not autodisabled. This option can be used with <code>-evacuate</code> but not with <code>-force</code> .
<code>-sys</code>	Stops HAD on the specified system.

About stopping VCS without the `-force` option

When VCS is stopped on a system without using the `-force` option, 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 493.

About stopping VCS with options other than the `-force` option

When VCS is stopped by options other than `-force` on a system with online service groups, the groups that run on the system are taken offline and remain offline. VCS indicates this by setting the attribute `IntentOnline` to 0. Use the option `-force` to enable service groups to continue being online while the VCS engine (HAD) is brought down and restarted. The value of the `IntentOnline` attribute remains unchanged after the VCS engine restarts.

About controlling the `hastop` behavior by using the `EngineShutdown` attribute

Use the `EngineShutdown` attribute to define VCS behavior when a user runs the `hastop` command.

Note: VCS does not consider this attribute when the `hastop` is issued with the following options: `-force` or `-local -evacuate -noautodisable`.

Configure one of the following values for the attribute depending on the desired functionality for the `hastop` command:

[Table 6-3](#) shows the engine shutdown values for the attribute.

Table 6-3 Engine shutdown values

EngineShutdown Value	Description
Enable	Process all hastop commands. This is the default behavior.
Disable	Reject all hastop commands.
DisableClusStop	Do not process the hastop -all command; process all other hastop commands.
PromptClusStop	Prompt for user confirmation before you run the <code>hastop -all</code> command; process all other hastop commands.
PromptLocal	Prompt for user confirmation before you run the <code>hastop -local</code> command; process all other hastop commands except <code>hastop -sys</code> command.
PromptAlways	Prompt for user confirmation before you run any <code>hastop</code> command.

Additional considerations for stopping VCS

Following are some additional considerations for stopping VCS:

- If you use 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 systems that run several applications, consider increasing the value of the `ShutdownTimeout` attribute.
- If you stop VCS on a system by using the `hastop` command, it autodisables each service group that includes the system in their `SystemList` attribute. VCS does not initiate online of the servicegroup when in an autodisable state. (This does not apply to systems that are powered off)
- If you use the `-evacuate` option, evacuation occurs before VCS is brought down. But when there are dependencies between the service groups while `-evacuate` command is issued, VCS rejects the command

About managing VCS configuration files

This section describes how to verify, back up, and restore VCS configuration files.

See [“About the main.cf file”](#) on page 50.

See [“About the types.cf file”](#) on page 55.

About 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.

Note: If UAC is enabled, run the program or commands in the “Run as administrator” mode even if the logged-on user belongs to the local administrators group. Alternatively, log on as an Administrator (default administrator account) to perform the tasks.

See [“Setting the configuration to read or write”](#) on page 107.

About 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.03Dec2001.17.59.04`. In addition, the previous version of any `.cf` file is saved with the suffix `.previous`; for example, `main.cf.previous`.

Verifying a configuration

Use `hacf` to verify (check syntax of) the `main.cf` and the type definition file, `types.cf`. VCS does not run if `hacf` detects errors in the configuration.

To verify a configuration

- ◆ Run the following command:

```
# hacf -verify config_directory
```

The variable `config_directory` refers to directories containing a `main.cf` file and any `.cf` files included in `main.cf`.

No error message and a return value of zero indicates that the syntax is legal.

Scheduling automatic backups for VCS configuration files

Configure the `BackupInterval` attribute to instruct VCS to create a back up of the configuration periodically. VCS backs up the `main.cf` and `types.cf` files as `main.cf.autobackup` and `types.cf.autobackup`, respectively.

To start periodic backups of VCS configuration files

- ◆ Set the cluster-level attribute BackupInterval to a non-zero value.

For example, to back up the configuration every 5 minutes, set BackupInterval to 5.

Example:

```
# haclus -display | grep BackupInterval

BackupInterval          0

# haconf -makerw

# haclus -modify BackupInterval 5

# haconf -dump -makero
```

Saving a configuration

When you save a configuration, VCS renames the file main.cf.autobackup to main.cf. VCS also save your running configuration to the file main.cf.autobackup.

If have not configured the BackupInterval attribute, VCS saves the running configuration.

See [“Scheduling automatic backups for VCS configuration files”](#) on page 106.

To save a configuration

- ◆ Run the following command

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

The option -makero sets the configuration to read-only.

Setting the configuration to read or write

This topic describes how to set the configuration to read/write.

To set the mode to read or write

- ◆ Type the following command:

```
# haconf -makerw
```

Displaying configuration files in the correct format

When you manually edit VCS configuration files (for example, the main.cf or types.cf file), you create formatting issues that prevent the files from being parsed correctly.

To display the configuration files in the correct format

- ◆ Run the following commands to display the configuration files in the correct format:

```
# hacf -cftocmd config  
# hacf -cmdtocf config
```

About managing VCS users from the command line

You can add, modify, and delete users on any system in the cluster, provided you have the privileges to do so.

If VCS is running in secure mode, specify fully-qualified user names, in the format `username@domain`. You cannot assign or change passwords for users when VCS is running in secure mode.

Note: User names provided in the `domain\username` or `username@domain.com` formats do not work.

The commands to add, modify, and delete a user must be executed only as root or administrator and only if the VCS configuration is in read/write mode.

See [“Setting the configuration to read or write”](#) on page 107.

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

Adding a user

Users in the category Cluster Guest cannot add users.

To add a user

- 1 Set the configuration to read/write mode:

```
# haconf -makerw
```

- 2 Add the user:

```
# hauser -add user [-priv <Administrator|Operator> [-group  
service_groups]]
```

- 3 Enter a password when prompted.
- 4 Reset the configuration to read-only:

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

To add a user with cluster administrator access

- ◆ Type the following command:

```
# hauser -add user -priv Administrator
```

To add a user with cluster operator access

- ◆ Type the following command:

```
# hauser -add user -priv Operator
```

To add a user with group administrator access

- ◆ Type the following command:

```
# hauser -add user -priv Administrator -group service_groups
```

To add a user with group operator access

- ◆ Type the following command:

```
# hauser -add user -priv Operator -group service_groups
```

To add a user on only one node with cluster administrator access

- 1 Set the configuration to read/write mode:

```
# haconf -makerw
```

- 2 Add the user:

```
# hauser -add user@node.domain -priv Administrator
```

For example,

```
# hauser -add user1@sys1.domain1.com -priv Administrator
```

- 3 Reset the configuration to read-only:

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

To add a user on only one node with group administrator access

- ◆ Type the following command:

```
# hauser -add user@node.domain -priv Administrator -group service_groups
```

Assigning and removing user privileges

The following procedure describes how to assign and remove user privileges:

To assign privileges to an administrator or operator

- ◆ Type the following command:

```
hauser -addpriv user Administrator|Operator
        [-group service_groups]
```

To remove privileges from an administrator or operator

- ◆ Type the following command:

```
hauser -delpriv user Administrator|Operator
        [-group service_groups]
```

To assign privileges to an OS user group

- ◆ Type the following command:

```
hauser -addpriv usergroup AdministratorGroup|OperatorGroup
        [-group service_groups]
```

To remove privileges from an OS user group

- ◆ Type the following command:

```
hauser -delpriv usergroup AdministratorGroup|OperatorGroup  
[-group service_groups]
```

Modifying a user

Users in the category Cluster Guest cannot modify users.

To modify a user

- 1 Set the configuration to read or write mode:

```
# haconf -makerw
```

- 2 Enter the following command to modify the user:

```
# hauser -update user
```

- 3 Enter a new password when prompted.

- 4 Reset the configuration to read-only:

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

Deleting a user

You can delete a user from the VCS configuration.

To delete a user

- 1 Set the configuration to read or write mode:

```
# haconf -makerw
```

- 2 For users with Administrator and Operator access, remove their privileges:

```
# hauser -delpriv user Administrator|Operator [-group  
service_groups]
```

- 3 Delete the user from the list of registered users:

```
# hauser -delete user
```

- 4 Reset the configuration to read-only:

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

Displaying a user

This topic describes how to display a list of users and their privileges.

To display a list of users

- ◆ Type the following command:

```
# hauser -list
```

To display the privileges of all users

- ◆ Type the following command:

```
# hauser -display
```

To display the privileges of a specific user

- ◆ Type the following command:

```
# hauser -display user
```

About querying VCS

VCS enables you to query various cluster objects, including resources, service groups, systems, resource types, agents, clusters, and sites. You may execute 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

This topic describes how to perform a query on service groups.

To display the state of a service group on a system

- ◆ Type the following command:

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

To display the resources for a service group

- ◆ Type the following command:

```
# hagrps -resources service_group
```

To display a list of a service group's dependencies

- ◆ Type the following command:

```
# hagrps -dep [service_group]
```

To display a service group on a system

- ◆ Type the following command:

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

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

To display the attributes of a system

- ◆ Type the following command:

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

Note that system names are case-sensitive.

To display the value of a service group attribute

- ◆ Type the following command:

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

To forecast the target system for failover

- ◆ Use the following command to view the target system to which a service group fails over during a fault:

```
# hagrps -forecast service_group [-policy failoverpolicy value] [-verbose]
```

The policy values can be any one of the following values: Priority, RoundRobin, Load or BiggestAvailable.

Note: You cannot use the `-forecast` option when the service group state is in transition. For example, VCS rejects the command if the service group is in transition to an online state or to an offline state.

The `-forecast` option is supported only for failover service groups. In case of offline failover service groups, VCS selects the target system based on the service group's failover policy.

The BiggestAvailable policy is applicable only when the service group attribute Load is defined and cluster attribute Statistics is enabled.

The actual service group FailOverPolicy can be configured as any policy, but the forecast is done as though FailOverPolicy is set to BiggestAvailable.

Querying resources

This topic describes how to perform a query on resources.

To display a resource's dependencies

- ◆ Enter the following command:

```
# hares -dep resource
```

Note: If you use this command to query atleast resource dependency, the dependency is displayed as a normal 1-to-1 dependency.

To display information about a resource

- ◆ Enter the following command:

```
# hares -display resource
```

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

To display resources of a service group

- ◆ Enter the following command:

```
# hares -display -group service_group
```

To display resources of a resource type

- ◆ Enter the following command:

```
# hares -display -type resource_type
```

To display resources on a system

- ◆ Enter the following command:

```
# hares -display -sys system
```

To display the value of a specific resource attribute

- ◆ Enter the following command:

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

To display atleast resource dependency

- ◆ Enter the following command:

```
# hares -dep -atleast resource
```

The following is an example of the command output:

```
#hares -dep -atleast
#Group  Parent  Child
g1      res1     res2,res3,res5,res6 min=2
g1      res7     res8
```

Querying resource types

This topic describes how to perform a query on resource types.

To display all resource types

- ◆ Type the following command:

```
# hatype -list
```

To display resources of a particular resource type

- ◆ Type the following command:

```
# hatype -resources resource_type
```

To display information about a resource type

- ◆ Type the following command:

```
# hatype -display resource_type
```

If *resource_type* is not specified, information regarding all types is displayed.

To display the value of a specific resource type attribute

- ◆ Type the following command:

```
# hatype -value resource_type attribute
```

Querying agents

[Table 6-4](#) lists the run-time status for the agents that the `haagent -display` command displays.

Table 6-4 Run-time status for the agents

Run-time status	Definition
Faults	Indicates the number of agent faults within one hour of the time the fault began 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).

To display the run-time status of an agent

- ◆ Type the following command:

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

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

To display the value of a specific agent attribute

- ◆ Type the following command:

```
# haagent -value agent attribute
```

Querying systems

This topic describes how to perform a query on systems.

To display a list of systems in the cluster

- ◆ Type the following command:

```
# hasys -list
```

To display information about each system

- ◆ Type the following command:

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

If you do not specify a system, the command displays attribute names and values for all systems.

To display the value of a specific system attribute

- ◆ Type the following command:

```
# hasys -value system attribute
```

To display system attributes related to resource utilization

- ◆ Use the following command to view the resource utilization of the following system attributes:

- Capacity
- HostAvailableForecast
- HostUtilization

```
# hasys -util system
```

The `-util` option is applicable only if you set the cluster attribute `Statistics` to `Enabled` and define at least one key in the cluster attribute `HostMeters`.

The command also indicates if the `HostUtilization`, and `HostAvailableForecast` values are stale.

Querying clusters

This topic describes how to perform a query on clusters.

To display the value of a specific cluster attribute

- ◆ Type the following command:

```
# haclus -value attribute
```

To display information about the cluster

- ◆ Type the following command:

```
# haclus -display
```

Querying status

This topic describes how to perform a query on status of service groups in the cluster.

Note: Run the `hastatus` command with the `-summary` option to prevent an incessant output of online state transitions. If the command is used without the option, it will repeatedly display online state transitions until it is interrupted by the command CTRL+C.

To display the status of all service groups in the cluster, including resources

- ◆ Type the following command:

```
# hastatus
```

To display the status of a particular service group, including its resources

- ◆ Type the following command:

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

If you do not specify a service group, the status of all service groups appears. The `-sound` option enables a bell to ring each time a resource faults.

To display the status of service groups and resources on specific systems

- ◆ Type the following command:

```
hastatus [-sound] -sys system_name  
[-sys system_name]...
```

To display the status of specific resources

- ◆ Type the following command:

```
hastatus [-sound] -resource resource_name
[-resource resource_name]...
```

To display the status of cluster faults, including faulted service groups, resources, systems, links, and agents

- ◆ Type the following command:

```
# hastatus -summary
```

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

- ◆ Type the following command:

```
# hamsg -help
```

To display the list of LDFs available on the current system

- ◆ Type the following command:

```
# hamsg -list
```

To display general LDF data

- ◆ Type the following command:

```
# 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:

```
C:\Program Files\VERITAS\Cluster Server\ldf
```

To display specific LDF data

- ◆ Type the following command:

```
# hamsg [-any] [-sev C|E|W|N|I]
[-otype VCS|RES|GRP|SYS|AGT]
```

```
[-oname object_name] [-cat category] [-msgid message_ID]  
[-path path_name] [-lang language] LDF_file
```

-any	Specifies <code>hamsg</code> return messages that match any of the specified query options.
-sev	Specifies <code>hamsg</code> return messages that match the specified message severity Critical, Error, Warning, Notice, or Information.
-otype	Specifies <code>hamsg</code> return messages that match the specified object type <ul style="list-style-type: none">■ VCS = general VCS messages■ RES = resource■ GRP = service group■ SYS = system■ AGT = agent
-oname	Specifies <code>hamsg</code> return messages that match the specified object name.
-cat	Specifies <code>hamsg</code> return messages that match the specified category. For example, the value 2 in the message id "V-16-2-13067"
-msgid	Specifies <code>hamsg</code> return messages that match the specified message ID. For example, the value 13067 the message id "V-16-2-13067"
-path	Specifies where <code>hamsg</code> looks for the specified LDF. If not specified, <code>hamsg</code> looks for files in the default directory: C:\Program Files\VERITAS\Cluster Server\ldf
-lang	Specifies the language in which to display messages. For example, the value <code>en</code> specifies English and <code>"ja"</code> specifies Japanese.

Using conditional statements to query VCS objects

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` returns all resources whose state begins with `FAULTED`.)

Multiple conditional statements can be used and imply `AND` logic.

You can only query attribute-value pairs that appear in the output of the command `hagrp -display`.

See [“Querying service groups”](#) on page 112.

To display the list of service groups whose values match a conditional statement

- ◆ Type the following command:

```
# hagrp -list [conditional_statement]
```

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

To display a list of resources whose values match a conditional statement

- ◆ Type the following command:

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

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

To display a list of agents whose values match a conditional statement

- ◆ Type the following command:

```
# haagent -list [conditional_statement]
```

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

About administering service groups

Administration of service groups includes tasks such as adding, deleting, or modifying service groups.

Adding and deleting service groups

This topic describes how to add or delete a service group.

To add a service group to your cluster

- ◆ Type the following command:

```
# 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.)

To delete a service group

- ◆ Type the following command:

```
# hagrps -delete service_group
```

Note that you cannot delete a service group until all of its resources are deleted.

Modifying service group attributes

This topic describes how to modify service group attributes.

To modify a service group attribute

- ◆ Type the following command:

```
# hagr -modify service_group attribute value [-sys system]
```

The variable *value* represents:

```
system_name1 priority1 system_name2 priority2
```

If the attribute that is being modified has local scope, you must specify the system on which to modify the attribute, except when modifying the attribute on the system from which you run the command.

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

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

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

```
# hagr -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
```

You cannot modify this attribute if resources have already been added to the service group.

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.

Modifying the SystemList attribute

You use the `hagr -modify` command to change a service group's existing system list, you can use the options `-modify`, `-add`, `-update`, `-delete`, or `-delete -keys`.

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
```

You must take the service group offline on the system that is being modified.

When you add a system to a service group's system list, the system must have been previously added to the cluster. When you use the command line, you can use the `hasys -add` command.

When you delete a system from a service group's system list, the service group must not be online on the system to be deleted.

If you attempt to change a service group's existing system list by using `hagrp -modify` without other options (such as `-add` or `-update`) the command fails.

Bringing service groups online

This topic describes how to bring service groups online.

To bring a service group online

- ◆ Type one of the following commands:

```
# hagrp -online service_group -sys system

# hagrp -online service_group -site [site_name]
```

To start a service group on a system and bring online only the resources already online on another system

- ◆ Type the following command:

```
# hagrp -online service_group -sys system
-checkpartial other_system
```

If the service group does not have resources online on the other system, the service group is brought online on the original system and the `checkpartial` option is ignored.

Note that the `checkpartial` option is used by the Preonline trigger during failover. When a service group that is configured with Preonline =1 fails over to another system (system 2), the only resources brought online on system 2 are those that were previously online on system 1 prior to failover.

To bring a service group and its associated child service groups online

- ◆ Type one of the following commands:

```
■ # hagrp -online -propagate service_group -sys system

■ # hagrp -online -propagate service_group -any

■ # hagrp -online -propagate service_group -site site
```

Note: See the man pages associated with the `hagrp` command for more information about the `-propagate` option.

Taking service groups offline

This topic describes how to take service groups offline.

To take a service group offline

- ◆ Type one of the following commands:

```
# hagrp -offline service_group -sys system
```

```
# hagrp -offline service_group -site site_name
```

To take a service group offline only if all resources are probed on the system

- ◆ Type the following command:

```
# hagrp -offline [-ifprobed] service_group -sys system
```

To take a service group and its associated parent service groups offline

- ◆ Type one of the following commands:

```
■ # hagrp -offline -propagate service_group -sys system
```

```
■ # hagrp -offline -propagate service_group -any
```

```
■ # hagrp -offline -propagate service_group -site site_name
```

Note: See the man pages associated with the `hagrp` command for more information about the `-propagate` option.

Switching service groups

The process of switching a service group involves taking it offline on its current system or site and bringing it online on another system or site.

To switch a service group from one system to another

- ◆ Type the following command:

```
# hagrps -switch service_group -to system  
  
# hagrps -switch service_group -site site_name
```

A service group can be switched only if it is fully or partially online. The `-switch` option is not supported for switching hybrid service groups across system zones.

Switch parallel global groups across clusters by using the following command:

```
# hagrps -switch service_group -any -clus remote_cluster
```

VCS brings the parallel service group online on all possible nodes in the remote cluster.

Freezing and unfreezing service groups

Freeze a service group to prevent it from failing over to another system. This freezing process stops all online and offline procedures on the service group.

Note that if the service group is in ONLINE state and if you freeze the service group, then the group continues to remain in ONLINE state.

Unfreeze a frozen service group to perform online or offline operations on the service group.

To freeze a service group (disable online, offline, and failover operations)

- ◆ Type the following command:

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

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

To unfreeze a service group (reenable online, offline, and failover operations)

- ◆ Type the following command:

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

Enabling and disabling priority based failover for a service group

This topic describes how to enable and disable priority based failover for a service group. Even though the priority based failover gets configured for all the service groups in a cluster, you must enable the feature at the cluster level.

To enable priority based failover of your service group

- Run the following command on the cluster:

```
# haclus -modify EnablePBF 1
```

This attribute enables Priority based failover for high priority service group. During a failover, VCS checks the load requirement for the high-priority service group and evaluates the best available target that meets the load requirement. If none of the available systems meet the load requirement, then VCS checks for any lower priority groups that can be offlined to meet the load requirement.

VCS performs the following checks, before failing over the high-priority service group:

- Check the best target system that meets the load requirement of the high-priority service group.
- If no system meets the load requirement, create evacuation plan for each target system to see which system will have the minimum disruption.
- Evacuation plan consists of a list of low priority service groups that need to be taken offline to meet the load requirement. Disruption factor is calculated for the system along with the evacuation plan. Disruption factor is based on the priority of the service groups that will be brought offline or evacuated. The following table shows the mapping of service group priority with the disruption factor:

Service group priority	Disruption factor
4	1
3	10
2	100
1	Will not be evacuated

- Choose the System with minimum disruption as the target system and execute evacuation plan. This initiates the offline of the low priority service groups. Plan is visible under attribute (Group::EvacList) . After all the groups in the evacuation plan are offline, initiate online of high priority service group.

To disable priority based failover of your service group:

- Run the following command on the service group:

```
# haclus -modify EnablePBF 0
```

Note: This attribute cannot be modified when evacuation of a group is in progress.

Enabling and disabling service groups

Enable a service group before you bring 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.

Disable a service group to prevent it from coming online. This process temporarily stops VCS from monitoring a service group on a system that is undergoing maintenance operations

To enable a service group

- ◆ Type the following command:

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

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

To disable a service group

- ◆ Type the following command:

```
# hagrps -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

- ◆ Type the following command:

```
# hagrps -enableresources service_group
```

To disable all resources in a service group

- ◆ Type the following command:

```
# hagrps -disableresources service_group
```

Agents do not monitor group resources if resources are disabled.

Clearing faulted resources in a service group

Clear a resource to remove a fault and make the resource available to go online.

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

- ◆ Type the following command:

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

Clearing a resource 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.

To clear resources in ADMIN_WAIT state in a service group

- ◆ Type the following command:

```
# hagr -clearadminwait [-fault] service_group -sys system
```

See “[Changing agent file paths and binaries](#)” on page 314.

Linking and unlinking service groups

This topic describes how to link service groups to create a dependency between them.

See “[About service group dependencies](#)” on page 335.

To link service groups

- ◆ Type the following command

```
# hagr -link parent_group child_group  
gd_category gd_location [gd_type]
```

<i>parent_group</i>	Name of the parent group
<i>child_group</i>	Name of the child group
<i>gd_category</i>	Category of group dependency (online/offline).
<i>gd_location</i>	The scope of dependency (local/global/remote/site).
<i>gd_type</i>	Type of group dependency (soft/firm/hard). Default is firm.

To unlink service groups

- ◆ Type the following command:

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

Administering agents

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

To start an agent

- ◆ Run the following command:

```
haagent -start agent -sys system
```

To stop an agent

- ◆ Run the following command:

```
haagent -stop agent [-force] -sys system
```

The `-force` option stops the agent even if the resources for the agent are online. Use the `-force` option when you want to upgrade an agent without taking its resources offline.

About administering resources

Administration of resources includes tasks such as adding, deleting, modifying, linking, unlinking, probing, and clearing resources, bringing resources online, and taking them offline.

About adding resources

When you add a resource, 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.
In an atleast resource set, the fault of a critical resource is tolerated until the minimum requirement associated with the resource set is met.
- **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: The addition of 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.

Adding resources

This topic describes how to add resources to a service group or remove resources from a service group.

To add a resource

- ◆ Type the following command:

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

The resource name must be unique throughout the cluster. The resource type must be defined in the configuration language. The resource belongs to the group *service_group*.

Deleting resources

This topic describes how to delete resources from a service group.

To delete a resource

- ◆ Type the following command:

```
hares -delete resource
```

Adding, deleting, and modifying resource attributes

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

To modify a new resource

- ◆ Type the following command:

```
hares -modify resource attribute value

hares -modify resource attribute value
[-sys system] [-wait [-time waittime]]
```

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

To set a new resource's Enabled attribute to 1

- ◆ Type the following command:

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

The agent managing the resource 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.

To add a resource attribute

- ◆ Type the following command:

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

The variable *value* is a -string (default), -integer, or -boolean.

The variable *dimension* is -scalar (default), -keylist, -assoc, 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

- ◆ Type the following command:

```
haattr -delete resource_type attribute
```

To add a static resource attribute

- ◆ Type the following command:

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

To delete a static resource attribute

- ◆ Type the following command:

```
haattr -delete -static resource_type static_attribute
```

To add a temporary resource attribute

- ◆ Type the following command:

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

To delete a temporary resource attribute

- ◆ Type the following command:

```
haattr -delete -temp resource_type attribute
```

To modify the default value of a resource attribute

- ◆ Type the following command:

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

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

Defining attributes as local

Localizing an attribute means that the attribute has a per-system value for each system listed in the group's SystemList. 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
```

Note that global attributes cannot be modified with the `hares -local` command.

[Table 6-5](#) lists the commands to be used to localize attributes depending on their dimension.

Table 6-5 Making VCS attributes local

Dimension	Task and Command
scalar	Replace a value: <pre>-modify [object] <i>attribute_name value</i> [-sys <i>system</i>]</pre>

Table 6-5 Making VCS attributes local (*continued*)

Dimension	Task and Command
vector	<ul style="list-style-type: none">■ Replace list of values: <code>-modify [object] attribute_name value [-sys system]</code>■ Add list of values to existing list: <code>-modify [object] attribute_name -add value [-sys system]</code>■ Update list with user-supplied values: <code>-modify [object] attribute_name -update entry_value ... [-sys system]</code>■ Delete all values in list (you cannot delete an individual element of a vector): <code>-modify [object] attribute_name -delete -keys [-sys system]</code>
keylist	<ul style="list-style-type: none">■ Replace list of keys (duplicate keys not allowed): <code>-modify [object] attribute_name value ... [-sys system]</code>■ Add keys to list (duplicate keys not allowed): <code>-modify [object] attribute_name -add value ... [-sys system]</code>■ Delete user-supplied keys from list: <code>-modify [object] attribute_name -delete key ... [-sys system]</code>■ Delete all keys from list: <code>-modify [object] attribute_name -delete -keys [-sys system]</code>

Table 6-5 Making VCS attributes local (*continued*)

Dimension	Task and Command
association	<ul style="list-style-type: none">■ 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> <p>Note: 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.</p> <pre>hagrp -modify group1 SystemList -delete sysa sysb [-sys system]</pre>

Linking and unlinking resources

Link resources to specify a dependency between them. A resource can have an unlimited number of parents and children. When you link 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.

To link resources

- ◆ Enter the following command:

```
hares -link parent_resource child_resource
```

The variable *parent_resource* depends on *child_resource* being online before going online itself. Conversely, *parent_resource* goes offline before *child_resource* goes offline.

For example, a NIC resource must be available before an IP resource can go online, so for resources IP1 of type IP and NIC1 of type NIC, specify the dependency as:

```
hares -link IP1 NIC1
```

To unlink resources

- ◆ Enter the following command:

```
hares -unlink parent_resource child_resource
```

Bringing resources online

This topic describes how to bring a resource online.

To bring a resource online

- ◆ Type the following command:

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

Taking resources offline

This topic describes how to take a resource offline.

To take a resource offline

- ◆ Type the following command:

```
hares -offline [-ignoreparent|parentprop] resource -sys system
```

The option `-ignoreparent` enables a resource to be taken offline even if its parent resources in the service group are online. This option does not work if taking the resources offline violates the group dependency.

To take a resource and its parent resources offline

- ◆ Type the following command:

```
hares -offline -parentprop resource -sys system
```

The command stops all parent resources in order before taking the specific resource offline.

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

- ◆ Type the following command:

```
hares -offprop [-ignoreparent] resource -sys system
```

As in the above command, the option `-ignoreparent` enables a resource to be taken offline even if its parent resources in the service group are online. This option does not work if taking the resources offline violates the group dependency.

Probing a resource

This topic describes how to probe a resource.

To prompt an agent to monitor a resource on a system

- ◆ Type the following command:

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

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

Clearing a resource

This topic describes how to clear a resource.

To clear a resource

- ◆ Type the following command:

Initiate a state change from RESOURCE_FAULTED to RESOURCE_OFFLINE:

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

Clearing a resource 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 [“Clearing faulted resources in a service group”](#) on page 129.

This command also clears the resource's parents. 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.

About administering resource types

Administration of resource types includes the following activities:

Adding, deleting, and modifying resource types

After you create a resource type, use the `haattr` command to add its attributes. By default, resource type information is stored in the `types.cf` configuration file.

To add a resource type

- ◆ Type the following command:

```
hatype -add resource_type
```

To delete a resource type

- ◆ Type the following command:

```
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

- ◆ Type the following command:

```
hatype -modify resource_type SourceFile ".\resource_type.cf"
```

The information regarding *resource_type* is stored in the file *config\resource_type.cf*.

An include line for *resource_type.cf* is added to the *main.cf* file.

Note: Make sure that the path to the SourceFile exists on all nodes before you run this command.

To set the value of static resource type attributes

- ◆ Type the following command for a scalar attribute:

```
hatype -modify resource_type attribute value
```

For more information, type:

```
hatype -help -modify
```

Overriding resource type static attributes

You can override some resource type static attributes and assign them resource-specific values. When a static attribute is overridden and the configuration is saved, the *main.cf* file includes a line in the resource definition for the static attribute and its overridden value.

To override a type's static attribute

- ◆ Type the following command:

```
hares -override resource static_attribute
```

To restore default settings to a type's static attribute

- ◆ Type the following command:

```
hares -undo_override resource static_attribute
```

Administering systems

Administration of systems includes tasks such as modifying system attributes, freezing or unfreezing systems, and running commands.

To modify a system's attributes

- ◆ Type the following command:

```
hasys -modify modify_options
```

Some attributes are internal to VCS and cannot be modified.

See [“About the -modify option”](#) on page 98.

To display the value of a system's node ID as defined in the llttab file

- ◆ Type the following command to display the value of a system's node ID as defined in the following file:

```
%VCS_HOME%\comms\llt\llttab.txt
```

```
hasys -nodeid [node_ID]
```

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

- ◆ Type the following command:

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

-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.
-------------	--

-evacuate	Fails over the system's active service groups to another system in the cluster before the freeze is enabled.
-----------	--

To unfreeze a frozen system (reenable online and switch of service groups)

- ◆ Type the following command:

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

To run a command on any system in a cluster

- ◆ Type the following command:

```
hacli -cmd command [-sys | -server system(s)]
```

Issues a command to be executed on the specified system(s). VCS must be running on the systems.

The use of the hacli command requires setting HcliUserLevel to at least COMMANDROOT. By default, the HcliUserLevel setting is NONE.

If the users do not want the root user on system A to enjoy root privileges on another system B, HcliUserLevel should remain set to NONE (the default) on system B.

You can specify multiple systems separated by a single space as arguments to the option -sys. If no system is specified, command runs on all systems in cluster with VCS in a RUNNING state. The command argument must be entered within double quotes if command includes any delimiters or options.

About administering clusters

Administration of clusters includes the following activities:

Retrieving version information

This topic describes how to retrieve information about the version of VCS running on the system.

To retrieve information about the VCS version on the system

- 1 Run one of the following commands to retrieve information about the engine version, the join version, the build date, and the PSTAMP.

```
had -version  
hastart -version
```

- 2 Run the following command to retrieve information about the engine version.

```
hastart -v
```

Using the `-wait` option in scripts that use VCS commands

The `-wait` option is for use in the scripts that use VCS commands to wait till an attribute value changes to the specified value. The option blocks the VCS command until the value of the specified attribute is changed or until the specified timeout expires. Specify the timeout in seconds.

The option can be used only with changes to scalar attributes.

The `-wait` option is supported with the following commands:

`haclus` `haclus -wait attribute value`
 `[-clus cluster] [-time timeout]`

Use the `-clus` option in a global cluster environment.

`hagrp` `hagrp -wait group attribute value`
 `[-clus cluster] [-sys system] [-time timeout]`

Use the `-sys` option when the scope of the attribute is local.

Use the `-clus` option in a global cluster environment.

`hares` `hares -wait resource attribute value`
 `[-clus cluster] [-sys system] [-time timeout]`

Use the `-sys` option when the scope of the attribute is local.

Use the `-clus` option in a global cluster environment.

`hasys` `hasys -wait system attribute value`
 `[-clus cluster] [-time timeout]`

Use the `-clus` option in a global cluster environment.

About administering simulated clusters from the command line

VCS Simulator is a tool to assist you in building and simulating cluster configurations. With VCS Simulator you can predict service group behavior during cluster or system faults, view state transitions, and designate and fine-tune various configuration parameters. This tool is especially useful when you evaluate complex, multi-node configurations. It is convenient in that you can design a specific configuration without test clusters or changes to existing configurations.

You can also fine-tune values for attributes that govern the rules of failover, such as Load and Capacity in a simulated environment. VCS Simulator enables you to simulate various configurations and provides the information that you need to make the right choices. It also enables simulating global clusters.

Configuring resources and applications in VCS

This chapter includes the following topics:

- [About configuring resources and applications](#)
- [About Virtual Business Services](#)
- [About Intelligent Resource Monitoring \(IMF\)](#)
- [About fast failover](#)
- [How VCS monitors storage components](#)
- [About storage configuration](#)
- [About configuring network resources](#)
- [About configuring file shares](#)
- [About configuring IIS sites](#)
- [About configuring services](#)
- [About configuring processes](#)
- [About configuring Microsoft Message Queuing \(MSMQ\)](#)
- [About configuring the infrastructure and support agents](#)
- [About configuring applications using the Application Configuration Wizard](#)
- [Configuring the service group in a non-shared storage environment](#)
- [About the VCS Application Manager utility](#)

- [About testing resource failover using virtual fire drills](#)
- [HA and DR configurations using InfoScale in AWS](#)
- [HA and DR configurations in Azure environment](#)

About configuring resources and applications

VCS monitors resources using agents. VCS detects the state of an application by continuously monitoring resources used by an application. If all resources required by the application are available, VCS declares the application as available.

Refer to the *Cluster Server Bundled Agents Reference Guide* for a description of the agents provided by VCS.

VCS provides configuration wizards to configure commonly-used resources. You can also use Cluster Manager (Java console) and the command line to configure resources.

Note: When modifying agent attributes from the Cluster Manager (Java console), use a single slash (\) to denote path names. When manually editing the configuration file `main.cf` directly, use double slashes (\\).

Configuring resources and applications in VCS involves the following tasks:

- Creating a service group comprising all resources required for the application and then configuring the resources.
For example, to configure a database in VCS, you must configure resources for the database and for the underlying shared storage and network. Configuring a resource involves defining values for its attributes. The resources must be logically grouped in a service group. When a resource faults, the entire service group fails over to another node.
- Assigning dependencies between resources.
For example, a MountV resource depends on a VMDg resource. Similarly, an IP resource depends on a NIC resource.
- Bringing the service group online.

Considerations for Windows Server systems

Consider the following items before configuring VCS resources and service groups on Windows Server systems:

- User Access Control (UAC)

If UAC is enabled, run the program or commands in the “Run as administrator” mode even if the logged-on user belongs to the local administrators group. Alternatively, log on as an Administrator (default administrator account) to perform the tasks.

For more information about UAC, refer to the Microsoft documentation.

- Windows Firewall

If you have configured the Windows firewall, ensure that the firewall settings allow access to the services and ports used by SFW HA.

Refer to the *Storage Foundation Installation and Upgrade Guide* for a detailed list of SFW HA services and ports used.

- Windows Server Core

If you want to configure VCS resources and service groups on Windows Server Core systems, you must manually add the required resources and configure the service groups. You can perform the steps either directly on the Windows Server Core system using the command line, or remotely using the Cluster Manager (Java console).

See [“About administering VCS from the command line”](#) on page 97.

Before configuring resources and service groups, review the resource types and the attribute definitions described in the *Cluster Server Bundled Agents Reference Guide*.

About Virtual Business Services

Virtual Business Services provide continuous high availability and reduce frequency and duration of service disruptions for multi-tier business applications running on heterogeneous operating systems and virtualization technologies. A Virtual Business Service represents the multi-tier application as a single consolidated entity and builds on the high availability and disaster recovery provided for the individual tiers by products such as Cluster Server and ApplicationHA. Additionally, a Virtual Business Service can also represent all the assets used by the service such as arrays, hosts, and file systems, though they are not migrated between server tiers. A Virtual Business Service provides a single consolidated entity that represents a multi-tier business service in its entirety. Application components that are managed by Cluster Server or ApplicationHA can be actively managed through a Virtual Business Service.

You can configure and manage Virtual Business Services created in Veritas InfoScale Operations Manager by using Veritas InfoScale Operations Manager Virtual Business Services. Besides providing all the functionality that was earlier available through Business Entity Operations Add-on, VBS provides the additional

ability to configure fault dependencies between the components of the multi-tier application.

Note: All the Application Entities that were created using Veritas Operations Manager Virtual Business Service Operations Add-on versions 3.1 and 4.0 are available as Virtual Business Services after you deploy the VBS in Veritas InfoScale Operations Manager 7.3.1. Veritas InfoScale Operations Manager 7.3.1 is a prerequisite for running Virtual Business Services.

Features of Virtual Business Services

You can use the VBS to perform the following tasks:

- Start Virtual Business Services from the Veritas InfoScale Operations Manager console. When a Virtual Business Service starts, its associated service groups are brought online.
- Stop Virtual Business Services from the Veritas InfoScale Operations Manager console. When a Virtual Business Service stops, its associated service groups are taken offline.

Applications that are under the control of Symantec ApplicationHA can be part of a Virtual Business Service. Symantec ApplicationHA enables starting, stopping, and monitoring of an application within a virtual machine.

If applications are hosted on VMware virtual machines, you can configure the virtual machines to automatically start or stop when you start or stop the Virtual Business Service.

- Establish service group relationships and set the order to bring service groups online and to take them offline. It ensures that the service groups from different clusters are brought online or taken offline in the correct order. This order is governed by the service group's relationships with other service groups, which are referred to as child service groups. Setting the correct order of service group dependency is critical to achieve business continuity and high availability.
- Establish service group relationships and specify the required reaction of an application component to a high availability event in an underlying tier.
- Manage the Virtual Business Service from Veritas InfoScale Operations Manager or from the clusters participating in the Virtual Business Service.
- Recover the entire Virtual Business Service to a remote site when a disaster occurs.

However, the following operations cannot be managed using VBS:

- The service group operations that are performed using the Veritas Cluster Server management console.

- The service group operations that are performed using the Veritas Cluster Server command-line interface.
- The service group operations that are performed using the Veritas Cluster Server Java console.
- VBS is not supported for composite Virtual Business Services. You can use it only for Virtual Business Services.

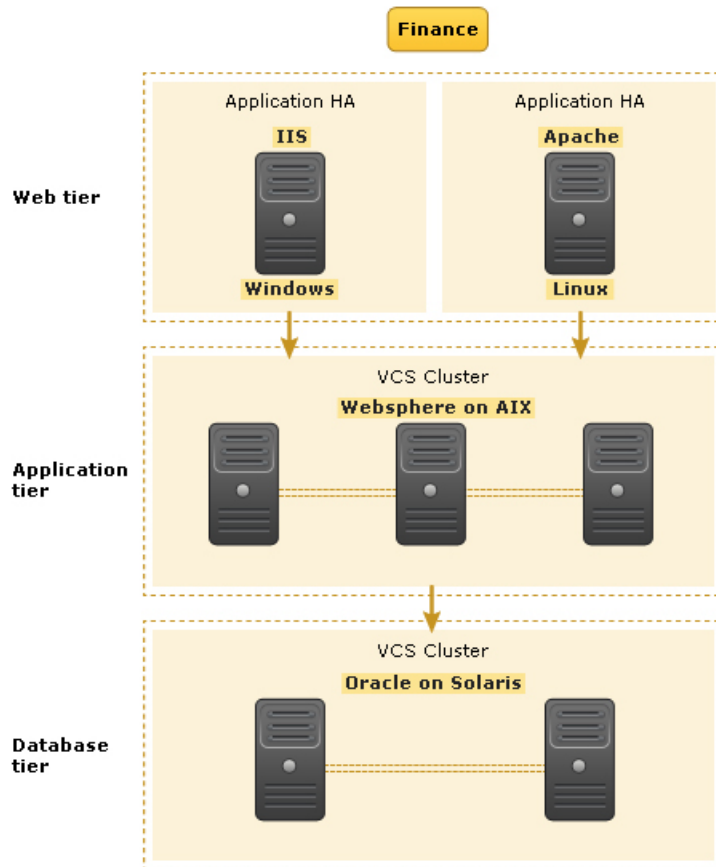
Note: You must install the VRTSvbs package on the cluster nodes to enable fault management and to administer the Virtual Business Service from the participating clusters.

Sample virtual business service configuration

This section provides a sample virtual business service configuration comprising a multi-tier application. [Figure 7-1](#) shows a Finance application that is dependent on components that run on three different operating systems and on three different clusters.

- Databases such as Oracle running on Solaris operating systems form the database tier.
- Middleware applications such as WebSphere running on AIX operating systems form the middle tier.
- Web applications such as Apache and IIS running on Windows and Linux virtual machines form the Web tier.

Each tier can have its own high availability mechanism. For example, you can use Cluster Server for the databases and middleware applications for the Web servers.

Figure 7-1 Sample virtual business service configuration

Each time you start the Finance business application, typically you need to bring the components online in the following order – Oracle database, WebSphere, Apache and IIS. In addition, you must bring the virtual machines online before you start the Web tier. To stop the Finance application, you must take the components offline in the reverse order. From the business perspective, the Finance service is unavailable if any of the tiers becomes unavailable.

When you configure the Finance application as a virtual business service, you can specify that the Oracle database must start first, followed by WebSphere and the Web servers. The reverse order automatically applies when you stop the virtual business service. When you start or stop the virtual business service, the components of the service are started or stopped in the defined order.

For more information about Virtual Business Services, refer to the *Virtual Business Service–Availability User's Guide*.

About Intelligent Resource Monitoring (IMF)

VCS traditionally uses a poll-based mechanism to detect the state of the configured applications and the underlying storage and network components. The agents retrieve the respective resource status during the monitor function. The monitor function is periodic and the frequency is defined by the resource type level attributes, `MonitorInterval` and `OfflineMonitorInterval`.

Intelligent Monitoring Framework (IMF) provides an alternative method for VCS to determine the resource status. IMF employs an event-based monitoring framework that is implemented using custom as well as native operating system-based notification mechanisms.

In poll-based monitoring, the resource state change detection is dependent on the monitor interval. Any state change that occurs immediately after a monitor cycle has completed is detected only in the next monitor cycle. This causes delays in fault detection. If the monitor interval attributes are set lower values, then in configurations with a large number of resources, poll-based monitoring may get CPU-intensive.

IMF uses an event-driven design that is asynchronous and provides instantaneous resource state change notifications. A resource state change event is quickly detected by VCS agents and then communicated to the VCS engine for further action. This improves the fault detection capability significantly allowing VCS to take corrective actions faster and that results in reduced service group failover times.

Note: The actual intelligent monitoring for a VCS resource starts only after two consecutive traditional monitor cycles have run and have returned the same state for that resource. So it takes some time before you see positive performance effect after enabling IMF.

The benefits of intelligent monitoring over poll-based monitoring are as follows:

- Instantaneous notification
Faster notification of resource state changes result in improved service group failover times.
- Reduction in system resource utilization
Reduced CPU utilization by VCS agent processes when number of resources being monitored is high. This provides significant performance benefits in terms of system resource utilization.

- Ability to monitor large number of resources
With reduced CPU consumption, IMF enables VCS to effectively monitor a large number of resources.

VCS changes to support IMF

IMF is an extension of the VCS agent framework. New IMF-related functions are added to the framework. The VCS agents can use these functions to register for IMF-based monitoring and communicate the resource state changes to the VCS high availability engine (HAD).

Agent Function	Description
imf_init	Initializes the agent interface for IMF-based monitoring. This function runs when the agent starts up.
imf_getnotification	<p>Gets notification about resource state changes. This function runs after the agent interface is initialized for IMF-based monitoring.</p> <p>During this function the agent continuously waits for an event and takes action on the resource upon notification.</p>
imf_register	<p>Agents use this function to register or unregister resource entities for IMF-based monitoring. For example, to register a process resource, the agent uses the process ID (PID) of the configured process for online monitoring of the process.</p> <p>This function runs for each resource after the resource has reported the same steady state (either online or offline) for two consecutive monitor cycles.</p>

Apart from the new agent functions, a new resource type level attribute, IMF, is introduced. The IMF attribute has three keys: Mode, MonitorFreq, RegisterRetryLimit. A combination of these keys determine whether or not an agent uses IMF-based monitoring for the corresponding resource type.

The default value of each key varies according to the agent that supports IMF. For information about bundled agents, refer to the *Cluster Server Bundled Agents Reference Guide*. For information about an application-specific agent, refer to the appropriate corresponding configuration guide.

Resource Type Attribute	Description
IMF	<p>Determines whether the IMF-aware agent must perform intelligent resource monitoring. You can also override the value of this attribute at resource-level.</p> <p>Type and dimension: integer-association</p> <p>This attribute includes the following keys:</p> <ul style="list-style-type: none"> Mode Define this attribute to enable or disable intelligent resource monitoring. This key takes the following values: <ul style="list-style-type: none"> 0 —Does not perform intelligent resource monitoring 1 —Performs intelligent resource monitoring for offline resources and poll-based monitoring for online resources 2 —Performs intelligent resource monitoring for online resources and poll-based monitoring for offline resources 3 —Performs intelligent resource monitoring for both online and for offline resources MonitorFreq This key value specifies the frequency at which the agent invokes the monitor agent function. The value of this key is an integer. After the resource is registered for IMF-based monitoring, the agent calls the monitor agent function as follows: <ul style="list-style-type: none"> For online resources: (MonitorFreq X MonitorInterval) number of seconds. For offline resources: (MonitorFreq X OfflineMonitorInterval) number of seconds. You can set this attribute to a non-zero value in cases where the agent requires to perform poll-based resource monitoring in addition to the intelligent resource monitoring. See the <i>Cluster Server Bundled Agents Reference Guide</i> for agent-specific recommendations. RegisterRetryLimit If you enable IMF-based monitoring, the agent runs the <code>imf_register</code> function to register the resource. The value of the <code>RegisterRetryLimit</code> key determines the number of times the agent must retry registration for a resource. If the agent cannot register the resource within the limit that is specified, then intelligent monitoring is disabled until the resource state changes or the value of the <code>Mode</code> key changes.

Resource Type Attribute	Description
IMFRegList	<p>An ordered list of attributes whose values are registered for IMF-based monitoring.</p> <p>Type and dimension: string-vector</p> <p>Default: Not applicable</p>

VCS agents that support IMF

The following agents support IMF-based monitoring:

- **GenericService, ServiceMonitor**
The agents trap the Windows service related events and takes appropriate action if a configured service stops or fails to respond.
- **IP, NIC**
These agents rely on the network and hardware events raised by the operating system. For example, an event is raised when an IP address becomes unavailable or when a network adapter is disabled.
- **MountV, Mount**
The agents use the PnP notifications generated by the operating system. In addition, the agent also uses custom notifications generated by Storage Foundation for Windows (SFW). For example, PnP notifications are generated for volume arrival or departure, volume failure, and file system notifications.
- **VMDg**
This agent relies on the disk group related PnP notifications raised by Storage Foundation for Windows (SFW). For example, SFW raises PnP notifications for disk group import and deport state change, and for disk group access state change (read-only, read/write).
- **Oracle, NetLsnr**
The agents trap the Windows service related events and takes appropriate action if a configured service stops or fails to respond.
- **Process, RegRep**
The Process agent supports IMF-based monitoring only when the resource is in the online state.
- **SQL Server, MSDTC**
This agent traps the Windows service related events and takes appropriate action if the configured SQL Server services stop or fail to respond.
- **IIS**

IMF-based monitoring support is limited only to monitoring the IIS services (FTP service, World Wide Web Publishing Service) that are necessary for the functioning of IIS. The agent traps the Windows service related events and takes appropriate action if a configured service stops or fails to respond. IMF is not used for monitoring the availability of the sites configured.

- Exchange2010DB

How IMF works

The following steps outline how IMF-based monitoring works:

1. After IMF is enabled for a resource, the corresponding VCS agent waits for the resource to report the same steady state (whether online or offline) for two consecutive monitor cycles and then registers the resource for IMF-based monitoring.
2. The VCS agent then registers itself for receiving specific custom or operating system specific event notifications.
3. If an event occurs, the agent determines the affected resource and then executes a monitor cycle for that resource. The monitor cycle determines the resource status and communicates it to the VCS engine (HAD). If the resource state is offline, the VCS engine (HAD) may initiate a failover depending on the configuration.
4. If the resource state remains the same, the agent moves to a wait state and then waits for the next event to occur.

How to enable IMF

IMF is enabled by default. However, you can manually enable it using the following steps.

To enable IMF

- 1 Change the VCS configuration to read/write mode.

Type the following at the command prompt:

```
haconf -makerw
```

- 2 Run the following command to enable intelligent resource monitoring:

- To enable intelligent monitoring of offline resources:

```
hatype -modify resource_type IMF -update Mode 1
```

- To enable intelligent monitoring of online resources:

```
hatype -modify resource_type IMF -update Mode 2
```

- To enable intelligent monitoring of both online and offline resources:

```
hatype -modify resource_type IMF -update Mode 3
```

- 3 Change values of MonitorFreq and the RegisterRetryLimit keys of the IMF attribute.
- 4 Save the VCS configuration.

Type the following at the command prompt:

```
haconf -dump -makero
```

How to disable IMF

IMF is enabled by default. Perform the following steps to disable it manually.

To disable IMF

- 1 Change the VCS configuration to read/write mode.

Type the following at the command prompt:

```
haconf -makerw
```

- 2 Run the following commands to disable intelligent resource monitoring:

- To disable intelligent resource monitoring for all the resources of a certain type, run the following command:

```
hatype -modify resource_type IMF -update Mode 0
```

- b. To disable intelligent resource monitoring for a specific resource, run the following commands:

```
hares -override resource_name IMF
```

```
hares -modify resource_name IMF -update Mode 0
```

- 3 Save the VCS configuration.

Type the following at the command prompt:

```
haconf -dump -makero
```

Recommended settings

Veritas recommends the following settings for faster failover and better performance.

Modify the MountV resource attributes

To reduce failover time, set the following MountV resource attributes:

- ListApplications = 0
This attribute defines whether the agent lists the applications that are accessing the volume while unmounting. Setting it to 0 avoids this enumeration and saves time during failover.
- ForceUnmount = ALL
Defines whether or not the agent unmounts the volume (either gracefully or forcibly) when it is being used by other applications. This graceful/forceful unmount takes additional time to close the Read-Only handles. Setting this to ALL saves time during failover.

Modify the attribute values for SQL Analysis Service and SQL Server Agent resources

Change the value of DelayAfterOnline and DelayAfterOffline attributes from the default to 25 for MS-Olap and SQL Server Agent resources.

The GenericService resources for MS-Olap service and SQL Server Agent service goes into an unknown state while trying to bring these services online. If the value of DelayAfterOnline and DelayAfterOffline attributes is set to 25 for MS-Olap and SQL Server Agent resources, then these resources come online without reporting unknown.

Modify the NumThreads attribute for MountV and VMDg

The VCS agent framework uses multithreading to allow multiple resource operations to run in parallel for the same type of resources. The NumThreads type level attribute value determines the number of threads for all the resource types.

- For large configurations, typically over 50 MountV resources, Veritas recommends that you set the MountV resource type NumThreads attribute value to 20.
- For large configurations, typically over 10 VMDg resources, Veritas recommends that you set the VMDg resource type NumThreads attribute value to 20.

About fast failover

Fast failover is a new feature that improves the failover time for the storage stack configured in a clustered environment. Fast failover includes several design changes and enhancements to the core SFW components. These changes provide significant reduction in the failover time taken by storage resources during service group failovers.

Fast failover integrates with the IMF feature to provide a significant performance improvement in SFW HA cluster environments. Fast failover requires a separate license. Fast failover appears as a selectable option in the SFW HA installer. Even though fast failover is installed and enabled, it will not work if the license does not support this feature.

Refer to the *Storage Foundation Administrator's Guide* for more information about fast failover.

VCS changes for fast failover

To support the fast failover feature, a new attribute, **FastFailOver**, is added to the VCS Volume Manager Diskgroup (VMDg) agent. This attribute decides whether or not a disk group is enabled for fast failover.

The **FastFailOver** attribute can take values of 1 and 0. The value 1 indicates that the agent enables fast failover for the configured disk group. The default value 0 indicates that fast failover is disabled for the disk group.

Refer to the *Cluster Server Bundled Agents Reference Guide* for more information about the VMDg agent.

Enabling fast failover for disk groups

Perform the following steps to enable fast failover for VMDg resources in service groups.

To enable the **FastFailover** attribute for a VMDg resource

- 1 In Cluster Manager (Java Console), select a service group with VMDg resource configured for it. Select the **Properties** tab from the right pane.
- 2 Scroll down to choose the **FastFailOver** attribute and click to edit the attribute value.
- 3 In the Edit Attribute dialog box, check the **FastFailOver** check box and then click **OK**.
- 4 Repeat these steps for every VMDg resource in the service groups.

How VCS monitors storage components

VCS provides specific agents that monitor storage components and ensure that the shared disks, disk groups, LUNs, volumes, and mounts are accessible on the system where the application is running. Separate agents are available for shared and non-shared storage and for third-party storage arrays such as NetApp filers. Your storage configuration determines which agent should be used in the high availability configuration.

For details on the various VCS storage agents, refer to the *Cluster Server Bundled Agents Reference Guide*.

Shared storage—if you use NetApp filers

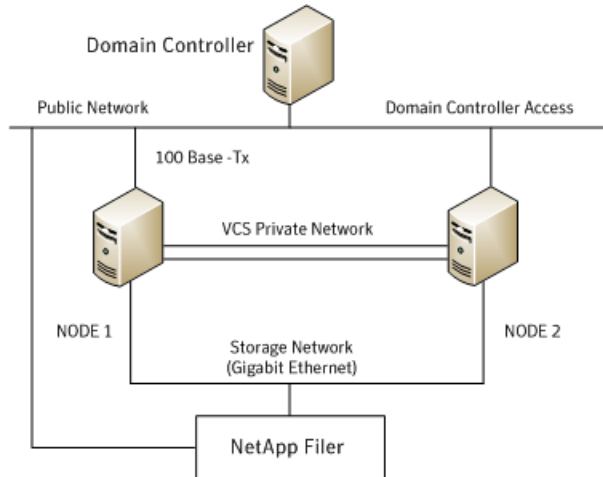
The VCS hardware replication agents for NetApp provide failover support and recovery in environments that employ NetApp filers for storage and NetApp SnapMirror for replication. The agents enable configuring NetApp filers over an iSCSI or Fibre Channel (FC) connection in a VCS cluster environment.

The VCS agents for NetApp are as follows:

- NetAppFiler
- NetAppSnapDrive
- NetAppSnapMirror

These agents monitor and manage the state of replicated filer devices and ensure that only one system has safe and exclusive access to the configured devices at a time. The agents can be used in local clusters, single VCS replicated data clusters, and multi-cluster environments that are set up using the VCS Global Cluster Option (GCO).

In a typical configuration, the agents are installed on each system in the cluster. The systems are connected to the NetApp filers through a dedicated (private) storage network. VCS cluster systems are physically attached to the NetApp filer via an ethernet cable supporting iSCSI or FC as the transport protocol.



VCS also provides agents for other third-party hardware arrays. For details on the supported arrays, refer to the product Software Compatibility List (SCL).

Shared storage—if you use SFW to manage cluster dynamic disk groups

The VCS MountV and VMDg agents are used to monitor shared storage that is managed using Storage Foundation (SFW). SFW manages storage by creating disk groups from physical disks. These disk groups are further divided into volumes that are mounted on the cluster systems.

The MountV agent monitors volumes residing on disk groups. The VMDg agent monitors cluster dynamic disk groups and is designed to work using SCSI reservations. Together the MountV and VMDg agents ensure that the shared cluster dynamic disk groups and volumes are available.

Shared storage—if you use Windows LDM to manage shared disks

The VCS Mount and DiskReservation (DiskRes) agents are used to monitor shared disks that are managed using Windows Logical Disk Management (LDM).

The Mount agent monitors basic disks and mount points and ensures that each system is able to access the volume or mount path in the same way. The DiskRes agent monitors shared disks and uses persistent reservation to ensure that only one system has exclusive access to the disks. During failovers, these agents ensure that the disks and volumes are deported and imported on the node where the application is running.

Non-shared storage—if you use SFW to manage dynamic disk groups

VCS introduces the Volume Manager Non-Shared Diskgroup (VMNSDg) agent to support local non-shared storage configurations that are managed using SFW. The VMNSDg agent works without SCSI reservations and is designed for locally attached storage devices that do not support SCSI.

The VMNSDg agent monitors and manages the import and deport of dynamic disk groups created on local storage. The only difference between the VMDg agent and the VMNSDg agent is that the VMDg agent is designed for shared cluster dynamic disk groups and uses SCSI reservations, whereas the VMNSDg agent supports only non-shared local dynamic disk groups and works without SCSI reservations.

The VMNSDg agent can be used to set up single node Replicated Data Clusters (RDC) or Disaster Recovery (DR) configurations with replication set up between the sites.

During a failover, the VCS MountV and VMNSDg agents deport the locally attached storage from the affected node and then import the locally attached storage of the target node. Replication ensures that the data is consistent and the application is up and running successfully.

Note: The VMNSDg agent does not support fast failover and Intelligent Monitoring Framework (IMF).

Non-shared storage—if you use Windows LDM to manage local disks

VCS introduces the NativeDisks agent to support local non-shared storage configurations managed using Windows LDM. The NativeDisks agent works without SCSI reservations and is designed for local storage that does not support SCSI.

Together with the Mount agent, the NativeDisks agent monitors and manages the import and deport of basic local disks on the system. The only difference between the DiskRes agent and the NativeDisks agent is that the DiskRes agent is designed for shared disks and uses SCSI reservations, whereas the NativeDisks agent supports only non-shared local disks and works without SCSI reservations.

Note: The NativeDisks agent does not support fast failover and Intelligent Monitoring Framework (IMF).

Non-shared storage—if you use VMware storage

VCS introduces the VMwareDisks agent to support storage configurations in a VMware virtual environment. The agent is platform independent and supports VMware Virtual Machine Disk (VMDK), Raw Device Mapping (RDM) disk files (virtual), and storage that is configured using Network File System (NFS). The VMwareDisks agent works without SCSI reservations and supports locally attached non-shared storage.

VMware features such as snapshots, vMotion, and DRS do not work when SCSI disks are shared between virtual machines. The VMwareDisks agent is designed to address this limitation. With this agent, the disks can now be attached to a single virtual machine at a time in the VCS cluster. On failover, along with the service group, the VMwareDisks agent moves the disks to the target virtual machine.

The VMwareDisks agent communicates with the host ESXi server to configure storage. This agent manages the disk attach and detach operations on a virtual machine in the VCS cluster. The agent is VMware HA aware. During failovers, the agent detaches the disk from one system and then attaches it to the system where the application is actively running. The VMwareDisks agent presents the virtual disks to the operating system. On Windows, the agent relies on the VMNSDg agent (in case of SFW-managed local storage) and the NativeDisks agent (in case of LDM-managed local storage) for initializing and managing the virtual disks. On Linux, the agent relies on the LVM and VxVM agents.

Note: The VMwareDisks agent does not support fast failover and Intelligent Monitoring Framework (IMF).

About storage configuration

Preview the following requirements before you configure shared storage:

- If your configuration uses shared disks and volumes managed by using Windows Logical Disk Manager (LDM), use the VCS DiskReservation and Mount agents. If you use LDM to manage non-shared local storage, use the VCS Mount and NativeDisks agents. For VMware storage environments, use the VMwareDisks agent in combination with the NativeDisks agent.
See [“About managing storage using Windows Logical Disk Manager”](#) on page 162.
- If your configuration uses shared volumes or Logical Unit Numbers (LUNs) managed in a Network Appliance storage environment, use the VCS NetAppSnapDrive and NetAppFiler agents. For NetApp replication, use the NetAppSnapMirror agent.

See [“About managing storage in a Network Appliance storage environment”](#) on page 166.

- If your configuration uses shared disks and volumes that are managed by using Storage Foundation for Windows (SFW), use the VCS MountV and VMDg agents. If you use SFW to manage non-shared local storage, use the VCS MountV and VMNSDg agents. For VMware storage environments, use the VMwareDisks agent in combination with the VMNSDg agent.

See [“About managing shared storage using Storage Foundation for Windows”](#) on page 168.

Before you configure the storage, review the resource type and the attribute definitions of these agents in the *Cluster Server Bundled Agents Reference Guide*.

About managing storage using Windows Logical Disk Manager

Before configuring storage, review the resource type and the attribute definitions of the Mount, DiskRes, NativeDisks, and VMwareDisks agents described in the *Cluster Server Bundled Agents Reference Guide*.

The following restrictions apply in this release:

- Mount, DiskRes, and NativeDisks agents are supported on VCS for Windows only. These agents are not supported if the storage is managed using Storage Foundation for Windows (SFW).
- If you are using shared storage, your storage devices must be configured to use SCSI-2 disk reservations. SCSI-3 is not supported.
SCSI support is not required if you are using non-shared storage.
- If NetApp SnapDrive is installed on the system, the service group configuration wizards are unable to detect LDM storage.
- LDM support is not applicable for Disaster Recovery configurations. Currently only HA configurations are supported.

Note the following prerequisites before configuring shared storage:

- Verify that the disk signature is the same on all systems sharing the disk.
- Install software drivers for the disk controller device identically on each system in the cluster. Verify that these driver services run successfully on all systems.
- Disable the option **Reset SCSI Bus at IC Initialization** from the SCSI Select utility.
- If using the agents in a Fibre Channel (FC) setup, enable target resets for the adapters.

- Verify that the device path to the disk is recognized by each system sharing the disk.
- Disable the write cache on the internal SCSI RAID controller.
- For each system, unassign the drive letter for the disk device path configured for mounting.
- If UAC is enabled, run the program or commands in the “Run as administrator” mode even if the logged-on user belongs to the local administrators group. Alternatively, log on as an Administrator (default administrator account) to perform the tasks.

Reserving disks (if you use Windows LDM)

Complete the following steps to reserve the disks on the node on which you are going to perform the application installation.

These steps are required only if you are configuring shared storage. Skip these steps for a non-shared storage configuration.

To reserve the disks

- 1 To display all the disks, type the following on the command line:

```
C:\>haval -scsitest /l
```

Make a note of the disk numbers (Disk# column in the table). You will need it in the next step.

- 2 To reserve a disk, type the following on the command line:

```
C:\>haval -scsitest /RES:<disk #>
```

For example, to reserve disk #4, type:

```
C:\>haval -scsitest /RES:4
```

Make a note of the disk number and the corresponding signature. You will require these details to identify and reserve the disks during installation and while configuring the service group, on additional nodes in the cluster.

Creating volumes (if you use Windows LDM)

Perform the following steps to create volumes.

To create volumes

- 1 Use the Windows Disk Management tool to verify that the disks are visible on the cluster nodes, and then create volumes on the disks.
- 2 In case of shared storage, after creating the required volumes on a node, release the reserved disks from that node.

See [“Releasing disks \(if you use Windows LDM\)”](#) on page 165.

- 3 3. In case of shared storage, rescan the disks on all the remaining nodes in the cluster.

Refer to Microsoft Windows documentation for more information about the Disk Management tool.

Mounting volumes (if you use Windows LDM)

Perform the following steps to mount volumes on a cluster node.

To mount a volume

- 1 Use the Windows Disk Management tool to mount the volumes that you created earlier.
- 2 After mounting the volumes on a cluster node, run the CHKDSK command and verify that there are no errors on the mounted volumes.
- 3 Make a note of the drive letters that you assign to the mounted volumes.

Use the same drive letters while mounting these volumes on the remaining cluster nodes.

Refer to Microsoft Windows documentation for more information about the CHKDSK command and the Disk Management tool.

Unassigning a drive letter

In case of a shared storage configuration, while installing an application on multiple nodes, you must first unassign drive letters and release the disks from one node, and then reserve the disks, mount the volumes using the same drive letters and then install the application on the failover node.

These steps are required only if you are configuring shared storage. Skip these steps for a non-shared storage configuration.

Note: You must run Disk Management on all systems each time you add a shared disk. This ensures each disk has a valid signature written to it, and that the device paths and symbolic links are updated.

Complete these steps to unassign the drive letters from a node.

To unassign drive letter

- 1 Log in as Administrator.
- 2 Open Disk Management. Type the following at the command prompt:


```
C:\> diskmgmt.msc
```
- 3 Right-click the partition or logical drive and click **Change Drive Letter and Path**.
- 4 In the **Change Drive Letter and Paths** dialog box, click the drive letter and click **Remove**.

Releasing disks (if you use Windows LDM)

Perform the following steps to release reserved disks from a cluster node.

These steps are required only if you are configuring shared storage. Skip these steps for a non-shared storage configuration.

To release disks

- 1 To display all the disks, type the following on the command line:

```
C:\>haval -scsitest /l
```

Make a note of the disk numbers (Disk# column in the table) of the disk that you wish to release. You will need it in the next step.

- 2 To release a reserved disk, type the following on the command line:

```
C:\>haval -scsitest /REL:<disk #>
```

For example, to release disk 4, type:

```
C:\>haval -scsitest /REL:4
```

Make a note of the disk number and the corresponding signature. You may require these details to identify and reserve the disks later.

Configuration tasks

This topic describes how to manually configure Mount and DiskRes resources in the cluster.

The same steps apply if you wish to configure any of the VCS agents manually. Refer to the application-specific agent or solutions guide for attribute details of the

application agents and the *VCS Bundled Agents Reference Guide* for details about the storage, network, services, and other infrastructure agents.

- 1 In your service group, create resources of type DiskReservation and Mount.
- 2 Configure the following required attributes for the following resources:

DiskReservation Resource
 - Signatures: An array specifying the signature of each SCSI disk. VCS provides the havol utility to retrieve disk signatures.

See [“The havol utility”](#) on page 504.
Mount Resource
 - MountPath: The drive letter or path to an empty NTFS folder that will be assigned to the volume being mounted.
 - PartitionNo: The partition on the disk configured for mounting.
 - Signature: A system-specified disk identifier. VCS provides the havol utility to retrieve the disk signature.
See [“The havol utility”](#) on page 504.
- 3 Link the Mount and DiskReservation resources such that the Mount resource depends on the DiskReservation resource.
- 4 Configure other resources in the service group, if required.
- 5 Bring the Mount resource online.

About managing storage in a Network Appliance storage environment

Network Appliance (NetApp) manages data by creating volumes on physical disks. These volumes can be divided further into Logical Unit Numbers (LUNs). The LUNs are accessible from the cluster nodes, provided the nodes have Microsoft iSCSI Initiator and Network Appliance SnapDrive installed. If you plan to use Fibre Channel (FC) for connecting the LUNs, ensure that you install the NetApp FCP Attach Kit on all the cluster nodes.

Perform the following tasks to create the required LUNs on the Network Appliance Filer and to make them accessible from cluster nodes:

- Create volumes on the Network Appliance Filer.
- Share the volumes.
- Create LUNs on the shared volumes.

Refer to Network Appliance documentation for instructions on performing these tasks.

Configuring Microsoft iSCSI Initiator

The Microsoft iSCSI initiator enables communication between Windows systems and Network Appliance Filers. The initiator uses the iSCSI protocol to present the filer volume as a local block device to the system.

To configure Microsoft iSCSI initiator on a Windows Server system

- 1 Start the Microsoft iSCSI initiator.
- 2 On the Discovery tab, click **Discover Portal**.
- 3 On the Discover Target Portal dialog box, specify the DNS name for the NetApp Filer and then click **OK**.
- 4 On the Target tab, click **Connect**.
- 5 On the Connect to Target dialog box, clear the **Add this connection to list of Favorite Targets** check box and then click **OK**.
- 6 On the Targets tab, verify that the newly added portal is listed under the **Select a target** box and the status appears as 'connected.' Then, click **OK**.

Connecting virtual disks to the cluster node

Once the virtual disks are created on the NetApp filer, they must be connected (if not connected already) to the cluster nodes using NetApp SnapDrive.

To connect virtual disks to the cluster node

- 1 On the cluster node where you want to connect the LUN, launch the Computer Management MMC from **Start > All Programs > Administrative Tools > Computer Management** or, on Windows Server 2012 operating systems, click **Administrative Tools** from the **Start** screen.
- 2 From the left pane, expand **Storage** and double-click **SnapDrive**.
- 3 Right-click **Disks** and then click **Connect Disk** to launch the Connect Disk wizard.
- 4 Click **Next** on the Welcome page.
- 5 Specify the path of the virtual disk that you wish to connect to the cluster node and then click **Next**.
- 6 Select **Dedicated** as the Virtual Disk Type and then click **Next**.
- 7 Click **Assign a Drive Letter** and then choose a drive letter from the drop-down list.
- 8 On the Select Initiator panel, specify the initiator(s) for the virtual disk and then click **Next**.

- 9 On the igroup Management Type panel, choose the option that allows SnapDrive to perform igroup management automatically and then click **Next**.
- 10 Click **Finish** to begin connecting the specified virtual disk to the cluster node.

Disconnecting virtual disks from the cluster nodes

Perform the following steps to disconnect the virtual disks from a cluster node.

To disconnect virtual disks

- 1 On the cluster node where you want to disconnect the LUNs, launch the Computer Management MMC from **Start > All Programs > Administrative Tools > Computer Management** or, on Windows Server 2012 operating systems, click **Administrative Tools** from the **Start** screen .
- 2 From the left pane, expand **Storage** and double-click **SnapDrive**.
- 3 Double-click **Disks** to see the LUNs that are connected to the node.
- 4 Right-click the LUN you want to disconnect and then click **Disconnect Disk**.
- 5 In the Disconnect Disk alert box, click **OK**.

About managing shared storage using Storage Foundation for Windows

Before configuring shared storage, review the resource type and the attribute definitions of the VMDg and the MountV agents in the *Cluster Server Bundled Agents Reference Guide*.

Note: If your storage devices are SCSI-3 compliant and you want to use SCSI-3 Persistent Group Reservations (PGR), you must enable SCSI-3 support using the Veritas Enterprise Administrator (VEA - Control Panel - System Settings). See the *Storage Foundation Administrator's Guide* for more information.

Using SFW with VCS

The following advanced features of Storage Foundation for Windows (SFW) require special consideration when used in a VCS environment:

- **Deporting Disk Groups**
SFW does not allow disk groups configured as VCS resources to be deported. They must be brought online or taken offline using VCS.
- **Dynamic Group Split and Join (DGSJ)**

SFW does not allow splitting a disk group configured as a VCS resource if the split operation causes a volume configured as a VCS resource to be part of the target group.

SFW does not allow a disk group configured as a VCS resource to be the source disk group in a join operation.

- **Deleting Volumes**

SFW does not allow deleting volumes configured as VCS resources.

- **Volume Snap Back**

If a volume formed as a result of a Prepare and Snap Shot operation is configured as a VCS resource, SFW does not allow Snap Back operations on the volume.

See the *Storage Foundation Administrator's Guide* for more information about these operations.

Before you configure shared storage using SFW

Following are the prerequisites for managing shared storage using SFW:

- Verify that SFW HA or VCS for Windows is installed on all cluster systems.
- If you configured Windows Firewall, add port 2148 to the Firewall Exceptions list.
For a detailed list of services and ports used by SFW HA, refer to the product installation guide.
- Configure the clustered disk group using Storage Foundation. Verify the disk group contains shared disks only.
- Disable the option **Reset SCSI Bus at IC Initialization** from the SCSI Select utility.
- Create a separate clustered disk group for each application to be clustered. Do not create a clustered disk group for more than one application. Configure all volumes or LUNs to be part of the VCS configuration and of the same service group.
- Assign a unique disk group name to each clustered disk group within a cluster.
- Ensure that the device path to the shared disk group or LUNs is recognized by all systems sharing the disk.

Configuring shared storage

This topic describes how to configure shared storage.

To configure shared storage

- 1 In your service group, create the following resources:

- For SFW HA, create resources of type VMDg and MountV.
- For VCS for Windows, create resources of type NetAppFiler and NetAppSnapDrive.

2 Configure the following required attributes for the respective resources:

VMDg resource	<p>DiskGroupName</p> <p>The name of the cluster disk group. Retrieve the name by running the command <code>vxdg list</code>, or by using the VMGetDrive utility.</p> <p>See “The vmgetdrive utility” on page 507.</p>
MountV resource	<ul style="list-style-type: none"> ■ MountPath: The drive letter or path to an empty NTFS folder that will be assigned to the volume being mounted. ■ VolumeName: The name of the volume to be mounted. For example, the name could be Raid1, Stripe2, Volume01, etc. Use the VMGetDrive utility to retrieve the volume name. See “The vmgetdrive utility” on page 507. ■ VMDGResName: The name of the Volume Manager Diskgroup (VMDg) resource on which the MountV resource depends.
NetAppFiler resource	<ul style="list-style-type: none"> ■ FilerName: DNS-resolvable name or IP address of the locally attached filer. ■ StorageIP: The private storage IP address of the filer.
NetAppSnapDrive resource	<ul style="list-style-type: none"> ■ FilerResName: Name of the VCS NetAppFiler-type resource in the service group. ■ VolumeName: Name of the volume containing the virtual disk. Define the volume name in the same case as on the filer. ■ ShareName: Name of the CIFS share containing the virtual disk. ■ LUN: Name of the LUN on the filer that is presented to the host for mounting. Define the LUN name in the same case as on the filer. ■ MountPath: Drive letter to be assigned to the virtual disk. ■ Initiator: Name of the iSCSI or FC initiator that the host uses to connect to the virtual disks on the filer. You can retrieve this from the Disk Management console.

3 Link the resources as follows:

- For SFW HA, link MountV and VMDg resources such that the MountV resource depends on the VMDg resource.

- For VCS for Windows, link NetAppSnapDrive and NetAppFiler resources such that the NetAppSnapDrive resource depends on the NetAppFiler resource.
- 4 Configure other resources in the service group, if required.
 - 5 Bring the MountV or the NetAppSnapDrive resource online.

Managing storage

Configure the storage disks to save the application data.

VMware virtualization manages the application data by storing it on SAN LUNs (RDM file), or creating virtual disks on a local or networked storage attached to the ESX host using iSCSI, network, or Fibre Channel. The virtual disks reside on a datastore or a raw disk that exists on the storage disks used.

For more information, refer to the VMware documentation.

The application monitoring configuration in a VMware environment requires you to use the RDM or VMDK disk formats. During a failover, these disks can be deported from a system and imported to another system.

Consider the following to manage the storage disks:

- Use a networked storage and create virtual disks on the datastores that are accessible to all the ESX servers that hosts the VCS cluster systems.
- In case of virtual disks, create non-shared virtual disks (Thick Provision Lazy Zeroed).
- Add the virtual disks to the virtual machine on which you want to start the configured application.
- Create volumes on the virtual disks.

Note: If your storage configuration involves NetApp filers that are directly connected to the systems using iSCSI initiator, you cannot configure application monitoring in a virtual environment with non-shared disks.

The following VCS storage agents are used to monitor the storage components involving non-shared storage:

- If the storage is managed using SFW, the MountV, VMNSDg, and VMwareDisks agents are used.
- If the storage is managed using LDM, the Mount, NativeDisks, and VMwareDisks agents are used.

Before configuring the storage, you can review the resource types and attribute definitions of these VCS storage agents. For details refer to the *Cluster Server Bundled Agents Reference Guide*.

About configuring network resources

When you configure your network resources in a VCS cluster, consider the following:

- For configuring the network components on your systems, use the NIC and IP agents.
- If your cluster systems use virtual computer names, use the Lanman agent.

About configuring IP addresses on the systems

Before configuring the network resources, review the resource type and the attribute definitions of the NIC and IP agents described in the *Cluster Server Bundled Agents Reference Guide*.

Before you configure IP addresses on the systems

Following are the prerequisites to configure IP addresses in the systems:

- Ensure that the NIC has the correct administrative IP address and subnet mask (for IPv4 addresses) or prefix length (for IPv6 addresses).
- If the NICs have built-in failover support, disable it. Refer to the documentation provided by the NIC vendor.
- Do not configure IP addresses added from the Control Panel.
- Verify that the virtual IP address to be assigned is unique and is not in use on the network.
- Disable DHCP on the NIC.

Disabling DHCP

This topic describes how to disable DHCP:

To disable DHCP

- 1 Open the Network Connections Control Panel.
- 2 Right-click the network connection and click **Properties**.
- 3 In the Properties dialog box for the respective local area connection, select the **General** tab, if not already selected.
- 4 Select **Internet Protocol (TCP/IP)** and click **Properties**.

- 5 Verify that the **Obtain an IP address automatically** option is not selected.
- 6 Specify values for **IP address**, **Subnet mask**, and **Default Gateway**, if not already specified.
- 7 Click **OK** and close the Control Panel.

Configuring IP addresses on the systems

This topic describes how to configure IP addresses on the systems.

To configure IP addresses

- 1 In your service group, create resources of type NIC and IP.

- 2 Configure the following required attributes for these resources:

NIC Resource

- **MACAddress**: The physical address of the NIC to be monitored. You can retrieve the physical addresses of NICs using the command `ipconfig -all`. Note that this attribute is always local.
- **UseConnectionStatus**: A flag that defines whether the NIC maintains its connection status.

IP Resource

- **Address**: The unique virtual IP address to be assigned to the NIC.
- **MACAddress**: The physical address of the NIC to which the virtual IP address is assigned. Note that this attribute is always local.
- **SubNetMask**: In case of an IPv4, the subnet mask associated with the IPv4 address.
- **Prefix**: In case of IPv6, the prefix associated with the IPv6 address.
The prefix is generally represented as:
`ipv6-address/prefix-length`
For example, `2001:db8:0:1::/64`.
- Ensure that the value of the attribute **UseConnectionStatus** is correct. This value is set to True by default, and indicates that all NICs maintain their connection status. If **UseConnectionStatus** is set to False, ensure that the NIC has an IP address assigned and that at least one host is listed in the attribute **PingHostList**.

- 3 Link the IP and NIC resources such that the IP resource depends on the NIC resource.

- 4 Configure other resources in the service group, if required.
- 5 Bring the IP resource online.

About configuring virtual computer names

Before configuring the agent, review the resource type and the attribute definitions of the Lanman agent described in the *Cluster Server Bundled Agents Reference Guide*.

Before you configure virtual computer names

Following are the prerequisites to configure virtual computer names:

- Remove static entries mapping the virtual name to the IP address from your WINS server.
- If using the agent to bind multiple IP addresses to a virtual computer name, make sure the IP addresses belong to different subnets.
- Make sure the VCS Helper domain user account has "Add workstations to domain" privilege enabled in the Active Directory.
- DNS scavenging affects virtual servers configured in VCS because the Lanman agent uses DDNS to map virtual names with IP addresses. If you use scavenging, then you must set the DNSRefreshInterval attribute for the Lanman agent. This will enable the Lanman agent to refresh the resource records on the DNS servers. See the Lanman agent description in the *Cluster Bundled Agents Reference Guide*.

Configuring virtual computer names

This topic describes how to configure virtual computer names:

To configure virtual computer names

- 1 In your service group, create resources of type NIC and IP.
See [“About configuring IP addresses on the systems”](#) on page 172.
- 2 Create a resource of type Lanman.
- 3 Configure the following required attributes for the resource:
 - VirtualName: The virtual computer name to be assigned to the server.
 - IPResName: The name of the IP resource (in case of IPv4) or the IPv6 resource (in case of IPv6) on which the Lanman resource depends. The IPResName attribute is not required if you have the MultiNet attribute set to 1.

- 4 Link the IP and NIC resources such that
 - the IP resource depends on the NIC resource, and
 - the Lanman resource depends on the IP resource.
- 5 Configure other resources in the service group, if required.
- 6 Bring the Lanman resource and other resources in the service group online.

About configuring file shares

VCS provides several ways to configure file shares, including the configuration wizard, Cluster Manager (Java Console), and the command line. This section provides instructions on how to use the File Share Configuration Wizard to configure file shares.

On Windows Server Core, you have to add the required resources and configure the service group manually. You can perform the steps either directly on the Server Core machine using VCS commands from the command line, or remotely using the Cluster Manager (Java Console).

See [“About administering VCS from the command line”](#) on page 97.

If you want to configure file shares manually, consider the following:

- To configure a shared directory, use the FileShare agent.
- To configure multiple directories, use the CompositeFileShare agent.
- If UAC is enabled, run the program or commands in the “Run as administrator” mode even if the logged-on user belongs to the local administrators group. Alternatively, log on as an Administrator (default administrator account) to perform the tasks.
- Before configuring the service group, review the agent resource types and the attribute definitions described in the *Cluster Server Bundled Agents Reference Guide*.

Before you configure a file share service group

Note the following prerequisites before you configure a file share service group:

- Verify that you have local administrator privileges on the system from where you run the wizard.
- If you have configured a firewall, add the required ports and services to the Firewall Exception list.

For a detailed list of services and ports used, refer to the product installation and upgrade guide.

- Verify that the VCS high availability engine, HAD, is running on the system from which you run the wizard.
- Verify that the directories to be shared reside on shared disks that are accessible from the nodes that will be part of the file share service group.
- If your storage is SCSI-3 compliant and you wish to use SCSI-3 persistent reservations, enable SCSI-3 support using Veritas Enterprise Administrator (VEA). VEA is available with SFW HA only.
- Mount the drives or LUNs containing the shared directories on the system where you run the wizard. Unmount the drives or LUNs from other systems in the cluster.

See [“About managing storage using Windows Logical Disk Manager”](#) on page 162.

See [“About managing storage in a Network Appliance storage environment”](#) on page 166.

See [“About managing shared storage using Storage Foundation for Windows”](#) on page 168.

- Verify that the Veritas Command Server service is running on all the systems in the cluster.
- If NetBIOS is disabled over TCP/IP, you must set the Lanman agent's DNSUpdateRequired attribute value to 1 (True).
You can modify the Lanman resource attribute value after configuring the service group.
- Verify that you have the following information ready. The wizard prompts you for these details:
 - A unique virtual computer name to be assigned to the file share server
This is the name that the clients use to access the file shares. The virtual name must not exceed 15 characters. If you specify a virtual computer name in lowercase letters, the name is converted to uppercase. For example, the name VCSServer is converted to VCSSERVER.
 - A unique virtual IP address to be assigned to the file share server
The virtual IP address is required only if you wish to configure an IPv4 address. In case of IPv6, the wizard prompts you to select the IPv6 network and automatically generates an IPv6 address that is valid and unique on the network. The wizard uses the prefix that is advertised by the router on the IPv6 network.

Note: Windows Server does not support accessing file shares using a virtual IP address. You can work around this restriction by using non-scoped file shares.

See [“Creating non-scoped file shares configured with VCS ”](#) on page 185.

See [“Making non-scoped file shares accessible while using virtual server name or IP address if NetBIOS and WINS are disabled”](#) on page 187.

- The list of directories to be shared.

You can add existing shares to the VCS configuration. However, you cannot add special shares (shares created by the operating system for administrative and system use). For example, you cannot add the shares ADMIN\$, print\$, IPC\$, and *DriveLetter\$* to the VCS configuration.

Configuring file shares using the wizard

The File Share Configuration Wizard enables you to create and modify file share service groups, making file shares highly available in a VCS cluster.

To configure file shares using the File Share Configuration Wizard

- 1 Start the File Share Configuration Wizard.
On Windows 2012 operating systems, use the **Apps** menu.
- 2 Review the information on the Welcome panel and click **Next**.
- 3 On the Wizard Options panel, click **Create service group** and then click **Next**.

- 4 On the Service Group Configuration panel, specify the following service group details:

Service Group Name	Type a name for the file share service group.
Group System List	<p>Specify the systems on which to configure the service group.</p> <p>To add systems to the service group's system list, select the systems in the Available Cluster Systems list and click the right arrow.</p> <p>To remove systems from the service group's system list, select the systems in the Systems in Priority Order list and click the left arrow.</p> <p>To change a system's priority in the service group's system list, select the system from the Systems in Priority Order and click the up and down arrow.</p> <p>System priority defines the order in which service groups are failed over to systems. The system at the top of the list has the highest priority, while the system at the bottom of the list has the lowest priority.</p>
Include selected systems in the service group's AutoStartList attribute	<p>To enable the service group to automatically come online on one of the systems, select this checkbox.</p> <p>See "Service group attributes" on page 546.</p>

Click **Next**.

- 5 On the FileShare Configuration panel, specify the following configuration information for the file share resources to be created.

Virtual Computer Name	Type a unique virtual computer name to be assigned to the file share server. This is the name that the clients use to access the file shares. The virtual name must not exceed 15 characters.
-----------------------	---

Path	<p>Click the field and either type the path of the directory to be shared or click the ellipsis button (...) to browse for a directory. The selected directories must meet the following conditions:</p> <ul style="list-style-type: none"> ■ The selected drive, the mount path, and the file path must not exist in the VCS configuration. ■ The directories to be shared must reside on shared, non-system drives. <p>The wizard validates the selected directory and displays an error message if the directory does not meet any of the conditions.</p>
Share Name	<p>If a selected directory is already shared, the Share Name column lists the names by which it is shared. You can select a listed share name to make an existing share highly available. You can also create a new share for the same directory by typing a new share name.</p>
Remove	<p>To remove a file share from the configuration, click to select the file share, and then click Remove.</p>
Configure NetApp SnapMirror Resource(s)	<p>This is applicable in case of VCS for Windows only.</p> <p>Check the Configure NetApp SnapMirror Resource(s) check box if you wish to set up a disaster recovery configuration.</p> <p>The SnapMirror resource is used to monitor replication between filers at the primary and the secondary site, in a disaster recovery configuration.</p> <p>Note that you must configure the SnapMirror resource only after you have configured the cluster at the secondary site.</p>

Click **Next**.

- 6 On the Share Permissions panel, specify the users for the file shares and assign permissions to them as follows:

Select the FileShare	From the drop-down list, select the file share with which to associate user permissions, or select the default All FileShares to set the same permissions for all file shares.
Select the Permission	From the drop-down list, select the permission to be associated with the user.
Select the User	Click the ellipsis button (...), select a user, and click OK .
Add	Click to add the specified user to the Selected Users list. By default, all selected users are given the READ_ACCESS permission.
Selected Users	Displays a list of selected users and the file share permissions. You can configure a maximum of 50 users for each file share. To configure more users, create a user group. To change the file share permission associated with a user, click a user name in the Selected Users list and then select the desired permission from the Select the Permission drop-down list.
Remove	To deny file share access to a user, click the user name in the Selected Users list and click Remove .

Click **Next**.

- 7 On the Share Properties panel, set the share properties for the file shares as follows:

Select the FileShare	From the drop-down list, select a file share whose properties you wish to set.
Enable access-based enumeration for this file share	Check the Enable access-based enumeration check box to enable the Windows access-based enumeration feature on the selected file share.

User Limit

Specify the number of users that are allowed access to the selected file share.

Choose from the following options:

- **Maximum allowed users:** Select this option to allow access to the maximum numbers of users allowed on Windows.
- **Allow this number of users:** Select this option and then type the number of users that you wish to grant access to the selected file share.
 If you type zero or a value greater than what Windows supports, access is granted to the maximum users allowed on Windows.

Enable cache

Check the **Enable cache** check box to enable local caching of the contents of the selected file share. Then, specify how the contents of the file share are available to users for offline access.

In the drop-down list select from the following caching options:

- **Manual caching of files and programs:** Only the files and programs specified by the user are available offline. This sets the FileShare resource attribute ClientCacheType to MANUAL.
- **Automatic caching of programs:** All the files and programs that the users access from the file share are available offline. This sets the FileShare resource attribute ClientCacheType to DOCS.
- **Optimized automatic caching of files and programs:** All the files and programs, including executables, are cached locally. The next time the user accesses the executable files, they are launched from the local cache. This sets the FileShare resource attribute ClientCacheType to PROGRAMS.

Hide share

Check the **Hide Share** check box to make the new share a hidden share.

Share all subfolder

Check the **Share all subfolders** check box to share the subdirectories.

Hide child shares

Check the **Hide child shares** check box to hide the shared subdirectories.

Apply these settings to

To apply the specified share properties to multiple file shares simultaneously, do the following:

- 1 Click the ellipsis button (...).
- 2 On the Copy Share Properties dialog box, select the file shares from the Available Shares list and click the right arrow to move them to the Selected Shares list.

Note that only those files shares that are not already shared are available for selection.

- 3 Click **OK**.

Note: This option is not visible if you are configuring only one share in the service group.

Click **Next**.

- 8 This is applicable in case of VCS for Windows only.

On the Initiator Selection panel, select the initiator for the virtual disk from the list of available initiators displayed for each cluster node, and then click **Next**.

If you are configuring multipath I/O (MPIO) over Fibre Channel (FC), you must select at least two FC initiators for each cluster node. Note that the node from which you run this wizard already has an initiator selected by default. This is the initiator that was specified when you connected the LUNs to this cluster node.

- 9 On the Network Configuration panel, specify information related to your network as follows:
 - Select **IPv4** to configure an IPv4 address for the virtual server.
 - In the **Virtual IP Address** field, type a unique virtual IPv4 address for the virtual server.
 - In the **Subnet Mask** field, type the subnet to which the virtual IPv4 address belongs.
 - Select **IPv6** to configure an IPv6 address for the virtual server. The IPv6 option is disabled if the network does not support IPv6.
 - Select the prefix from the drop-down list. The wizard uses the prefix and automatically generates an IPv6 address that is valid and unique on the network.
 - For each system in the cluster, select the public network adapter name. This field displays the TCP/IP enabled adapters on a system, including the private network adapters, if applicable. To view the adapters associated

with a system, click the **Adapter Display Name** field and click the arrow. Verify that you select the adapters assigned to the public network, not the private.

- Click **Advanced Settings** to specify additional details for the Lanman resource.

On the Lanman Advanced Configuration dialog box, do the following:

- Check **Active Directory Update required** check box to enable the Lanman resource to update the Active Directory with the virtual name. This sets the Lanman agent attributes ADUpdateRequired and ADCriticalForOnline to true.
- In the **Organizational Unit** field, type the distinguished name of the Organizational Unit for the virtual server in the format
CN=containername,DC=domainname,DC=com.
 To browse for an Organizational Unit, click the ellipsis button (...) and search using the Windows Find Organization Units dialog box.
 By default, the Lanman resource adds the virtual server to the default container "Computers."
- Click **OK**.
 The user account for VCS Helper service must have adequate privileges on the specified container to create and update computer accounts.

Click **Next**.

- 10 On the Summary panel, review the service group configuration; the following service group details are displayed:

Resources	<p>Displays a list of configured resources. The wizard assigns unique names to resources. Change the names of resource, if required.</p> <p>To edit a resource name, select the resource name and either click it or press the F2 key. Edit the resource name and then press the Enter key to confirm the changes. To cancel editing a resource name, press the Esc key.</p>
Attributes	<p>Displays the attributes and their configured values, for a resource selected in the Resources list.</p>

Click **Next**.

- 11 Click **Yes** on the dialog that appears, informing you that the wizard will run commands to modify the service group configuration.
- 12 On the completion panel, check **Bring the service group online** check box if you want to bring the service group online on the local system, and then click **Finish**.

Modifying a file share service group using the wizard

The File Share Configuration Wizard enables you to modify a file share service group.

Consider the following before you modify file share service groups using the wizard:

- If the file share service group is online, you must run the wizard from a node on which the service group is online. You can then use the wizard to add resources to and remove them from the configuration. You cannot change attributes of resources that are online.
- To change the resource attributes, you must take the service group offline. However, the MountV and VMDg (in case of SFW HA), Mount and DiskRes (in case of Windows LDM), and NetAppSnapDrive and NetAppFile (in case of VCS for Windows) resources for the service group should be online on the node where you run the wizard and offline on all other nodes.
- If you are running the wizard to remove a node from the service group's system list, do not run the wizard on the node being removed.
- If the service group contains resources that were not part of the default service group configuration, then modifying the service group may change those resources. You may then have to manually restore the settings of those resources later.
- After configuring a file share if you move the shared directory to a new location, then while reconfiguring the file share service group, the wizard fails to delete the storage resources configured for the existing file share.

The wizard successfully creates a new file share resource and the corresponding storage resources, but fails to remove the older storage resources from the service group.

In such cases, you can either remove the stale storage resources manually, or delete the file share service group and run the wizard again to recreate the service group.

To modify a file share service group using the wizard

- 1 Start the File Share Configuration Wizard on a node on which the file share service group is online.

On Windows 2012 operating systems, use the **Apps** menu.
- 2 Review the information on the Welcome panel and click **Next**.
- 3 On the Wizard Options panel, click **Modify service group**, select the service group to be modified, and click **Next**.
- 4 Follow the wizard instructions and make desired modifications to the service group configuration.

See [“About configuring file shares”](#) on page 175.

Deleting a file share service group using the wizard

This topic describes steps to delete a file share service group using the wizard.

To delete a file share service group using the wizard

- 1 Start the File Share Configuration Wizard on a system configured to host the file share service group.

On Windows 2012 operating systems, use the **Apps** menu.
- 2 Review the information in the Welcome panel and click **Next**.
- 3 On the Wizard Options panel, click **Delete service group**, select the service group to be deleted, and click **Next**.
- 4 On the Service Group Summary panel, click **Next**. A message appears informing you that the wizard will run commands to delete the service group. Click **Yes** to delete the service group.
- 5 Click **Finish**.

Creating non-scoped file shares configured with VCS

File shares configured with VCS on Windows Server are accessible only using the virtual server name (Lanman resource). These file shares are not accessible using the IP address.

The FileShare agent is enhanced to address this issue. The FileShare agent behavior can be controlled using the following registry key:

```
HKLM\SOFTWARE\VERITAS\VCS\BundledAgents\
```

```
Lanman\virtualName\DisableServerNameScoping
```

Set the `DisableServerNameScoping` key to have the FileShare agent support non-scoped file shares.

You must create this registry key manually.

Note: Incorrectly editing the registry may severely damage your system. Back up the registry before making changes.

To configure the `DisableServerNameScoping` registry key

- 1 To open the Registry Editor, press **Window+R** on the desktop (opens the Run dialog box), type `regedit`, and then click **OK**.
- 2 In the registry tree (on the left), navigate to the following location:

```
HKLM\SOFTWARE\VERITAS\VCS\BundledAgents
```

- 3 Click **Edit > New > Key** and create a key by the name **Lanman**, if it does not exist already.
- 4 Select the **Lanman** key and click **Edit > New > Key** and create a key by the name ***virtualName***.

Here, *virtualName* should be the virtual computer name assigned to the file share server. This is the `VirtualName` attribute of the Lanman resource in the file share service group.

The newly created registry key should look like this:

```
HKLM\SOFTWARE\VERITAS\VCS\BundledAgents\  
Lanman\virtualName
```

- 5 Select the key that you created in step 4 (***virtualName***) and add a **DWORD** type of value.

The value name should be `DisableServerNameScoping` and value data should be 1.

The value 1 indicates that the FileShare and Lanman agents support non-scoped file shares on Windows Server systems.

- 6 If there are multiple file share service groups to be used in the non-scoped mode, repeat steps 4 and 5 for each Lanman resource that is configured in the file share service group.
- 7 Save and exit the Registry Editor.

You must create this key only for Lanman resources that are part of VCS file share service groups. Configuring this key for Lanman resources that are part of other VCS service groups may result in unexpected behavior.

Making non-scoped file shares accessible while using virtual server name or IP address if NetBIOS and WINS are disabled

The VCS FileShare agent depends on NetBIOS or DNS to resolve the virtual name. If NetBIOS and WINS are disabled or the DNS is not updated, the agent is unable to resolve the virtual name.

This may typically occur when the file share service groups are configured to use localized IP addresses. When the service group is switched or failed over, the virtual name to IP address mapping changes. In such a case if WINS database or the DNS are not updated, the agent is unable to resolve the virtual name. As a result the FileShare resources fault and the shares become inaccessible.

The following message appears in the agent log:

```
VCS INFO V-16-10051-10530 FileShare:servicegroupname:online:  
Failed to access the network path (\\virtualName)
```

The FileShare agent is enhanced to address this issue. The FileShare agent behavior can be controlled using the following registry key:

```
HKLM\SOFTWARE\VERITAS\VCS\BundledAgents\  
\Lanman\virtualName\DisableStrictVirtualNameCheck
```

Set the `DisableStrictVirtualNameCheck` key to have the FileShare agent make the file shares accessible irrespective of whether or not the virtual name is resolvable. In case the virtual name is not resolvable, the file shares are accessible using the virtual IP.

You must create this registry key manually.

Note: Incorrectly editing the registry may severely damage your system. Back up the registry before making changes.

To configure the `DisableStrictVirtualNameCheck` registry key

- 1 To open the Registry Editor, press **Window+R** on the desktop (opens the Run dialog box), type `regedit`, and then click **OK**.
- 2 In the registry tree (on the left), navigate to the following location:

```
HKLM\SOFTWARE\VERITAS\VCS\BundledAgents
```

- 3 Click **Edit > New > Key** and create a key by the name **Lanman**, if it does not exist already.
- 4 Select the **Lanman** key and click **Edit > New > Key** and create a key by the name **virtualName**.

Here, *virtualName* should be the virtual computer name assigned to the file share server. This is the VirtualName attribute of the Lanman resource in the file share service group.

The newly created registry key should look like this:

```
HKLM\SOFTWARE\VERITAS\VCS\BundledAgents\  
Lanman\virtualName
```

- 5 Select the key that you created in step 4 (**virtualName**) and add a DWORD type of value.

The value name should be DisableStrictVirtualNameCheck and value data should be 1.
- 6 If there are multiple file share service groups to be used in the non-scoped mode, repeat steps 4 and 5 for each Lanman resource that is configured in the file share service group.
- 7 Save and exit the Registry Editor.

You must create this key only for Lanman resources that are part of VCS file share service groups. Configuring this key for Lanman resources that are part of other VCS service groups may result in unexpected behavior.

About configuring IIS sites

When configuring the IIS agent to monitor a Web site, you can monitor associated application pools in the following ways:

- Configure a single resource to monitor both, the Web site and the associated application pools. In this case you define options to monitor associated application pools within the same resource.
- Configure separate resources to monitor IIS site and associated application pools. In this case you configure a resource to monitor the IIS site only and configure additional resources to monitor specific application pools.

VCS provides several ways to configure the agent, including the configuration wizard, Cluster Manager (Java console), and the command line. This section provides instructions on how to use the wizard to configure monitoring for IIS.

To configure the VCS IIS agent on Windows Server Core, first install IIS on Windows Server Core systems in the order specified. Then, manually add the required resources and configure the service group. You can perform the manual configuration steps either directly on the Server Core machine using VCS commands from the command line, or remotely using the Cluster Manager (Java console).

If UAC is enabled, run the program or commands in the “Run as administrator” mode even if the logged-on user belongs to the local administrators group. Alternatively, log on as an Administrator (default administrator account) to perform the tasks.

Review the IIS agent’s resource type definition and attribute descriptions in the *Cluster Server Bundled Agents Reference Guide*. Also, review the sample configurations and resource dependency graphs.

Refer to the following for more information:

See [“Installing IIS on Windows Server Core”](#) on page 191.

See [“Before you configure an IIS service group”](#) on page 189.

See [“Configuring an IIS service group using the wizard”](#) on page 193.

See [“About administering VCS from the command line”](#) on page 97.

Before you configure an IIS service group

Note the following prerequisites before you configure an IIS service group:

- Verify IIS is installed and configured identically on all nodes hosting the service group. Verify that the sites to be monitored are on shared storage.
- For IIS 8.0 (on Windows Server 2012 or Windows Server 2012 R2), and IIS 10.0 (on Windows Server 2016) you must install the following role services:

- IIS 6 Metabase Compatibility
 - IIS 6 WMI Compatibility or the IIS Management Scripts and Tools
- Only one of these role services is required.

These options are available under Management Tools on the Role Services page of the Add Roles Wizard.

If IIS 6 Metabase Compatibility role is installed, the WMI 6 Provider is used. If IIS Management Scripts and Tools role is installed, the WMI 7 Provider is used. If both the roles are installed, the WMI 7 Provider is used.

These components are required for the IIS agent to function on Windows Server.

- For Windows Server Core editions, you must install IIS in the specified order. See [“Installing IIS on Windows Server Core”](#) on page 191.

- If IIS configuration is using IPv6 addresses, then you must install the IIS Management Scripts and Tools role service.
 IPv6 requires WMI 7 Provider that is part of the IIS Management Scripts and Tools role.
- If you are configuring FTP sites that use IPv6 addresses, ensure that the IPv6 address entry (IP Address column in Site Bindings dialog) is enclosed in square brackets. The VCS IIS Configuration Wizard requires this format to correctly configure the FTP site in the cluster.
 See [“Fixing the IPv6 address configuration for FTP sites”](#) on page 191.
- Do not use the IIS agent to configure SMTP and NNTP sites if you have Microsoft Exchange installed.
- Change the default home directory path for all IIS sites to monitored to a location on the shared storage. See the IIS documentation for instructions.
- Verify that the port numbers assigned to IIS sites are not used by other programs.
- Synchronize the IIS configuration on all nodes hosting the service group.
 See [“About configuring IIS sites”](#) on page 188.
- Verify that you have local administrator privileges on the system from where you run the wizard.
- If you have configured a firewall, add the required ports and services to the Firewall Exception list.
 For a detailed list of services and ports used refer to the product installation and upgrade guide.
- Verify that the VCS engine, HAD, is running on the node from which you run the wizard.
- Mount the drives or LUNs containing the shared directories on the node from which you run the wizard. Unmount the drives or LUNs from other nodes in the cluster.
 See [“About managing storage using Windows Logical Disk Manager”](#) on page 162.
 See [“About managing storage in a Network Appliance storage environment”](#) on page 166.
 See [“About managing shared storage using Storage Foundation for Windows”](#) on page 168.
- If your storage is SCSI-3 compliant and you wish to use SCSI-3 persistent reservations, enable SCSI-3 support using Veritas Enterprise Administrator (VEA).
- Keep the following information ready. The wizard prompts you for these details:
 - IIS sites to be monitored

- Application pools associated with each site
- Port numbers associated with each site
- Virtual IP addresses and computer names associated with the sites
The virtual IP addresses and the virtual computer names must have forward and reverse entries in the DNS.

Fixing the IPv6 address configuration for FTP sites

When you add an FTP site using the Add FTP Site wizard, the IPv6 address is not enclosed in brackets by default. The VCS IIS Configuration Wizard requires the IPv6 addresses enclosed in square brackets format to correctly configure the FTP site in the cluster.

1. From the IIS Manager, right-click the FTP site name and click **Bindings**.
2. In the Site Bindings dialog box, select the FTP site and click **Edit**.
3. In the Edit Site Binding dialog box, type square brackets around the IPv6 address displayed in the IP address field.

For example, the IPv6 address should display as

```
[2001:Db8:0:10:828:1871:cd8:5c0f].
```

4. Click **OK** and then click **Close**.

Installing IIS on Windows Server Core

On Windows Server Core, you must install IIS in the order specified in this procedure.

To install IIS on Windows Server Core

1 Type the following at the command prompt:

```
C:\>start /w pkgmgr
/iu:IIS-WebServerRole;IIS-WebServer;IIS-CommonHttpFeatures;
IIS-StaticContent;IIS-DefaultDocument;IIS-DirectoryBrowsing;
IIS-HttpErrors;IIS-HttpRedirect;IIS-ApplicationDevelopment;
IIS-ASP;IIS-CGI;IIS-ISAPIExtensions;IIS-ISAPIFilter;
IIS-ServerSideIncludes;IIS-HealthAndDiagnostics;
IIS-HttpLogging;IIS-LoggingLibraries;IIS-RequestMonitor;
IIS-HttpTracing;IIS-CustomLogging;IIS-ODBCLogging;IIS-Security;
IIS-BasicAuthentication;IIS-WindowsAuthentication;
IIS-DigestAuthentication;
IIS-ClientCertificateMappingAuthentication;
IIS-IISCertificateMappingAuthentication;
IIS-URLAuthorization;IIS-RequestFiltering;IIS-IPSecurity;
IIS-Performance;IIS-HttpCompressionStatic;
IIS-HttpCompressionDynamic;IIS-WebServerManagementTools;
IIS-ManagementScriptingTools;IIS-IIS6ManagementCompatibility;
IIS-Metabase;IIS-WMICompatibility;IIS-LegacyScripts;
IIS-FTPPublishingService;WAS-WindowsActivationService;
IIS-FTPPublishingService;IIS-FTPServer
```

2 Verify that all the components specified in the earlier step have successfully installed. Type the following at the command prompt:

```
C:\>notepad C:\windows\logs\cbs\cbd.log
```

This opens the log file, cbd.log, in the Notepad text editor.

3 Check the entries in the log file, cbd.log. The last log entry should resemble the following:

```
Info CBS Pkgmgr: return code: 0x0
```

This message indicates that all the components are installed successfully.

- 4 Run the `oclist` command to verify that the following components are installed:

IIS-WebServerRole; IIS-WebServer; IIS-IIS6ManagementCompatibility;
IIS-Metabase; IIS-WMICompatibility; IIS-FTTPublishingService;
WAS-WindowsActivationService; IIS-FTTPublishingService; IIS-FTPService

Type the following at the command prompt:

```
C:\>oclist
```

- 5 Repeat the steps on all the nodes where you want to configure the IIS service group.

Configuring an IIS service group using the wizard

The IIS Configuration Wizard enables you to create and modify IIS service groups, making sites highly available in VCS cluster.

The wizard creates one resource for each IIS site and its associated application pools; the wizard does not create resources that monitor only application pools.

To configure an IIS service group using the wizard

- 1 Start the IIS Configuration Wizard.
On Windows 2012 operating systems, use the **Apps** menu.
- 2 Review the information on the Welcome panel and click **Next**.
- 3 On the Wizard Options panel, click **Create service group** and click **Next**.

- On the Service Group Configuration panel, specify the service group details and then click **Next**.

Specify the following details:

Service Group Name	Type a name for the IIS service group.
Available Cluster Systems	<p>Select the systems on which to configure the service group and click the right arrow to move the systems to the service group's system list.</p> <p>To remove a system from the service group's system list, click the system in the Systems in Priority Order box and click the left arrow.</p> <p>To change a system's priority in the service group's system list, click the system from the Systems in Priority Order and click the up and down arrows.</p> <p>System priority defines the order in which service groups are failed over to systems. The system at the top of the list has the highest priority while the system at the bottom of the list has the lowest priority.</p>
Include selected systems in the service group's AutoStartList attribute	<p>To enable the service group to automatically come online on one of the systems, select this checkbox.</p> <p>See "Service group attributes" on page 546.</p>

- 5 On the Configure IIS Sites panel, add and remove sites from the service group, configure IP addresses, ports, and virtual computer names, optionally choose to configure NetApp SnapMirror resources and then click **Next**.

Specify the following details:

Add	Check the check box corresponding to the site to be configured in VCS.
IP	<p>Verify or type the virtual IP address for each site to be configured.</p> <p>Make sure that each virtual IP address is associated with only one virtual computer name and vice-versa.</p>
Port	Type the port number for each site to be configured.
Virtual Name	Type a virtual name for the selected site. Each virtual name can be associated with only one virtual IP address at a time.
Configure NetApp SnapMirror Resource(s)	<p>This is applicable with VCS for Windows only.</p> <p>Check the Configure NetApp SnapMirror Resource(s) check box if you want to set up a disaster recovery configuration. The SnapMirror resource is used to monitor replication between filers at the primary and the secondary site, in a disaster recovery configuration.</p> <p>Note that you must configure the SnapMirror resource only after you have configured the cluster at the secondary site.</p>

- 6 On the Network Configuration panel, specify information related to the virtual IP addresses and then click **Next**.

Specify the following details:

IP Address	Displays the virtual IP addresses. The wizard groups systems by the virtual IP addresses associated with the systems.
Subnet Mask	<p>If the virtual IP is an IPv4 address, verify or type the subnet mask associated with each virtual IPv4 address.</p> <p>If the virtual IP is an IPv6 address, verify or type the associated IPv6 prefix. The prefix is generally represented in the following format: <code>ipv6-address/prefix-length</code>.</p> <p>For example:</p> <p><code>2001:db8:0:1:::/64</code></p>
Adapter Name	Select the public adapter associated with the virtual IP address on each system.

- 7 This is applicable with VCS for Windows only.

On the Initiator Selection panel, select the initiator for the virtual disk from the list of available initiators displayed for each cluster node, and then click **Next**.

If you are configuring multiPath I/O (MPIO) over Fibre Channel (FC), you must select at least two FC initiators for each cluster node. Note that the node from which you run this wizard already has an initiator selected by default. This is the initiator that was specified when you connected the LUNs to this cluster node.

- 8 On the Application Pool Configuration panel, select the monitoring options for application pools associated with each site and then click **Next**.

Specify the following details:

Site Name	Displays the site names.
-----------	--------------------------

AppPoolMon For each site, select the monitoring options from the AppPoolMon list.

Choose from the following options from the drop-down list:

- **NONE**—The agent does not monitor the application pool associated with the site.
- **DEFAULT**—Starts and monitors the root application pool associated with the site.
- **ALL**—Starts all application pools associated with the site and monitors root application pool.

- 9 On the Service Group Summary panel, review the service group configuration and then click **Next**.

The following service group details are visible:

Resources	<p>Displays a list of configured resources. The wizard assigns unique names to resources. Change the names of resource, if required.</p> <p>To edit a resource name, select the resource name and either click it or press the F2 key. Edit the resource name and then press the Enter key to confirm the changes. To cancel editing a resource name, press the Esc key.</p>
Attributes	<p>Displays the attributes and their configured values, for a resource selected in the Resources list.</p>

- 10 Click **Yes** on the dialog that prompts you that the wizard will run commands to modify the service group configuration.
- 11 In the completion dialog box, check **Bring the service group online** if you want to bring the service group online on the local system, and then click **Finish**.

Modifying an IIS service group using the wizard

The IIS Configuration Wizard enables you to modify an IIS service group.

Consider the following before you modify an IIS service group:

- If the IIS service group is online, you must run the wizard from a node on which the service group is online. You can then use the wizard to add resources to and remove them from the configuration. You cannot change attributes of resources that are online.
- To change the resource attributes, you must take the service group offline. However, the MountV and VMDg (in case of SFW HA), Mount and DiskRes (in

case of Windows LDM), and NetAppSnapDrive and NetAppFiler (in case of VCS for Windows) resources for the service group should be online on the node where you run the wizard and offline on all other nodes.

- If you are running the wizard to remove a node from the service group's system list, do not run the wizard on the node being removed.
- If the service group contains resources that were not part of the default service group configuration, then modifying the service group may change those resources. You may then have to manually restore the settings of those resources later.

To modify the IIS service group

- 1 Start the IIS Configuration Wizard.
On Windows 2012 operating systems, use the **Apps** menu.
- 2 Review the information on the Welcome panel and click **Next**.
- 3 On the Wizard Options panel, click **Modify service group**, select the service group to be modified, and click **Next**.
- 4 Follow the wizard instructions and make the modifications that you want to the service group configuration.

See [“Configuring an IIS service group using the wizard”](#) on page 193.

Deleting an IIS service group using the wizard

This topic describes steps to delete an IIS service group using the configuration wizard.

To delete the IIS service group

- 1 Start the IIS Configuration Wizard on a system configured to host the IIS service group.
On Windows 2012 operating systems, use the **Apps** menu.
- 2 Review the information in the Welcome panel and click **Next**.
- 3 In the Wizard Options panel, click **Delete service group**, select the service group to be deleted, and click **Next**.
- 4 In the Service Group Summary panel, click **Next**. When the message appears that informs you that the wizard will run commands to delete the service group, click **Yes** to delete the service group.
- 5 Click **Finish**.

About configuring services

Use the GenericService and ServiceMonitor agents to configure services in a VCS cluster.

Consider the following before you proceed:

- To start, stop, and monitor a service, use the GenericService agent.
- To monitor a service, use the ServiceMonitor agent.

About configuring a service using the GenericService agent

The GenericService agent starts, stops, and monitors services. Before configuring the service group, review the resource types and attribute definitions of the GenericService agent, described in the *Cluster Server Bundled Agents Reference Guide*.

You can configure the GenericService agent manually, as described below, or by using the Application Configuration Wizard.

See [“About configuring applications using the Application Configuration Wizard”](#) on page 222.

On Windows Server Core systems, you have to add the required resources and configure the service group manually. You can perform the steps either directly on the Server Core machine using VCS commands from the command line, or remotely using the Cluster Manager (Java console).

See [“About administering VCS from the command line”](#) on page 97.

Before you configure a service using the GenericService agent

Note the following prerequisites before you configure a service using the GenericService agent:

- For the service that you want to configure, change the startup type of the service to **Manual** on all the nodes that will be part of the service group.
- Ensure that the service is stopped on all the nodes that will be part of the service group.
- If monitoring the service in a user-context, configure the service to start in the context of the specified user account. Make sure the check box **Allow service to interact with desktop** is cleared.

Changing a service startup type

Perform these steps to change the startup type of a service to manual.

To change a service startup type to Manual

- 1 Open the Windows Services Control Manager.
- 2 Right-click the service and click **Properties**.
- 3 In the **Properties** dialog box, click the **General** tab.
- 4 From the **Startup Type** list, select **Manual**.
- 5 Click **OK**.
- 6 Close the **Services** Control Manager.

Configuring a service to run in a user context

Perform the following steps to start a service in a user context.

To configure a service to start in a user-context

- 1 Open the Services Control Manager.
- 2 Right-click the service and click **Properties**.
- 3 In the **Properties** dialog box, click the **LogOn** tab.
- 4 Click **This Account**.
- 5 Click **Browse** to browse existing user accounts.
- 6 In the **Select User** dialog box, click the user in whose context you want to run the service and click **OK**.
- 7 Enter the password for the selected user.
- 8 Click **OK** and close the **Services** Control Manager.

Configuring a service using the GenericService agent

This topic describes how to manually configure a service using the GenericService agent.

To configure a service using the GenericService agent

- 1 In your service group, create a resource of type GenericService.
- 2 Configure the following required attribute for the GenericService resource:
 ServiceName: The name of the service to be monitored, as displayed in the Windows Service Control Manager console.
- 3 Configure the following optional attributes for the GenericService resource, if required:

- **UserAccount:** A valid user account in whose context the service will be monitored. User name can be of the form `username@domain.com` or `domain.com\username`. If you do not specify a value for this attribute, then the user account of the service in the SCM is ignored. To monitor service under built-in accounts, you must provide explicit values.

For example:

- **User Account="LocalSystem", "Local Service", or "Network Service".**
Domain="NT Authority".
- **Password:** The password for the user account.
- **Domain:** The domain name to which the user specified in the UserAccount attribute belongs.

- 4 Configure other resources in the service group, if required.
- 5 Bring the GenericService resource, and other resources in the service group, online.

About configuring a service using the ServiceMonitor agent

The ServiceMonitor agent monitors a service or starts a script that monitors a service. Before configuring the service group, review the resource types and attribute definitions of the agent, described in the *Cluster Server Bundled Agents Reference Guide*.

You can configure the agent manually, as described below, or by using the Application Configuration Wizard.

See [“About configuring applications using the Application Configuration Wizard”](#) on page 222.

Before you configure a service using the ServiceMonitor agent

Note the following prerequisites before you configure a service using the ServiceMonitor agent:

- If using the agent to start a script, copy the script locally on each node in the cluster.
- If using the agent to monitor a service, start the service in the context of the LocalSystem account or in the context of the user account specified in the configuration.
- Verify that the user in whose context the service or script needs to be started, exists as a domain user or LocalSystem user.

Configuring a service using the ServiceMonitor agent

This topic describes how to manually configure a service using the ServiceMonitor agent.

To configure a service using the ServiceMonitor agent

- 1 In your service group, create a resource of type ServiceMonitor.
- 2 Configure the following required attribute for the ServiceMonitor resource:
 - ServiceOrScriptName: The name of the service to be monitored using the Service Control Manager (SCM). When monitoring the service through a user defined script, specify the complete path of the script, including any command-line arguments.

When monitoring a service through a user-defined script, specify the following attribute values:

- MonitorService: A flag that defines whether the agent monitors a service using the SCM or starts a script to monitor a service. If the flag is set to 1, the agent monitors a service specified by the attribute ServiceOrScriptName. If the flag is set to 0 the agent starts a script specified by the attribute ServiceOrScriptName. Default is 1.
 - MonitorProgTimeout: The maximum wait time, in seconds, for the agent to receive a return value from the monitor script. This attribute is ignored if the MonitorService flag is set to 1. Default is 30 seconds.
- 3 Configure other resources in the service group, if required.
 - 4 Bring the ServiceMonitor resource, and other resources in the service group, online.

About configuring processes

Before configuring a Process resource, review the resource types and attribute definitions of the agent, described in the *Cluster Server Bundled Agents Reference Guide*.

You can configure a Process resource either manually, as described below, or by using the Application Configuration Wizard.

See [“About configuring applications using the Application Configuration Wizard”](#) on page 222.

On Windows Server Core systems, you have to add the required resources and configure the service group manually. You can perform the steps either directly on

the Server Core machine using the VCS commands, or remotely using the Cluster Manager (Java console).

See [“About administering VCS from the command line”](#) on page 97.

Before you configure processes

Note the following prerequisites before you configure processes:

- The executables configured as the start, stop, and monitor programs must reside on local drives.
- When defining the StartProgram, StopProgram, or MonitorProgram attributes, enclose the path of the executable file in double quotes. *Do not enclose arguments in double quotes.* For example, specify the StartProgram attribute in the following format:

```
StartProgram = "executable_pathname" arguments
```

Configuring processes using the Process agent

Complete the following steps to manually configure processes using the Process agent.

To configure a process

- 1 In your service group, create a resource of type Process.
- 2 Configure the following required attribute for the Process resource:
 - **StartProgram:** The process to be monitored by the agent. You must specify the complete path of the executable, its file extension, and command-line arguments, if any. If you define the start program as a script to launch another program, you must specify the monitor program in the configuration file.

If you define the start program as a script (a perl script, or a vbs script), the start program should be the program that interprets the script (perl.exe, or cscript.exe) and the script itself should be passed as an argument.
- 3 Configure the following optional attributes, if required:
 - **StartupDirectory:** The startup directory for the process indicated by the StartProgram attribute.
 - **MonitorProgram:** A program that monitors the process specified as the start program. You must specify the complete path of the executable, its file extension, and command-line arguments, if any. If you do not specify a

value for this attribute, VCS monitors the start program. However, if the start program is a script to launch another program, you must specify a monitor program.

- MonitorProgramTimeout: The maximum wait time, in seconds, for the agent to receive a return value from the monitor routine. This attribute is ignored if the monitor program is not specified.
- 4 Configure other resources in the service group, if required.
 - 5 Bring the Process resource, and other resources in the service group, online.

About configuring Microsoft Message Queuing (MSMQ)

VCS provides several ways to configure a Microsoft Message Queuing (MSMQ) service group that include the MSMQ Configuration Wizard, the Enterprise Vault Cluster Setup Wizard, Cluster Manager (Java Console), Web Console, and the command line.

To create an MSMQ service group from the Cluster Manager (Java Console), you can use the MSMQ service group configuration template, MSMQVMGroup (for SFW), or MSMQNetAppGroup (for NetApp). These templates are installed at `%vcs_home%\templates` directory.

Here, `%vcs_home%` is the default product installation directory, typically `C:\Program Files\Veritas\Cluster Server`.

Launch the Service Group Configuration Wizard (Tools > Configuration Wizard) from the Cluster Manager (Java Console) and use these templates to configure the respective application service groups.

The following topics describe how to configure an MSMQ service group using the MSMQ Configuration Wizard. You can use the MSMQ Configuration Wizard to configure a service group for MSMQ that is installed in Active Directory or in Workgroup mode. Veritas recommends that you use the MSMQ Configuration Wizard to create the MSMQ resource and other resources that it depends upon.

Make sure that you review the resource types and attribute definitions of the MSMQ agent in the *Cluster Server Bundled Agents Reference Guide*.

Note: Cluster support for MSMQ triggers is not available in this release. In domain mode MSMQ installation, if Routing Support is selected while installing MSMQ, it is not supported.

Before you configure the MSMQ service group

Note the following prerequisites before you configure the MSMQ service group:

- If you use VCS for Windows (not SFW HA) to make the MSMQ service highly available, reboot the system after installing VCS. Make sure that you do so before running the MSMQ Configuration Wizard to create a service group. Otherwise, the clustered MSMQ service fails to initialize, and therefore, the MSMQ resource fails to come online.
- Remove the Message Queuing Triggers service if it is already installed. Cluster support for MSMQ Triggers service is not available in this release.
- Create volumes or LUNs for the MSMQ data and registry replication information (RegRep) and then mount or connect the volumes or LUNs on the node where you run the wizard.
 You can use a single volume for both MSMQ data and registry information. Veritas recommends that you use separate volumes for these components.
 See [“About managing storage using Windows Logical Disk Manager”](#) on page 162.
 See [“About managing storage in a Network Appliance storage environment”](#) on page 166.
 See [“About managing shared storage using Storage Foundation for Windows”](#) on page 168.
- Create directories for MSMQ data and registry information on the mounted volumes. For example, if `x:` is the volume, then `x:\MSMQ\Storage` can be the storage path for MSMQ.
- If MSMQ is integrated with Windows Active Directory (AD), then ensure that the value of the Lanman resource attributes `ADUpdateRequired` and `ADCriticalForOnline` is set to 1, after the service group is configured.

Note: You may receive an error when you try to read messages from a remote public queue in Microsoft Message Queuing. See article [889860](#) in the Microsoft Knowledge Base for more information. To overcome this problem, set the value of the Lanman resource attributes `DNSUpdateRequired` and `DNSCriticalForOnline` to 1.

- Verify that all the existing services that are dependent on the default MSMQ service are in the STOPPED state.
- If MSMQ is installed in Domain Mode, perform the following steps before you bring the MSMQ resource online for the first time:
 - First, bring the Lanman resource online in the service group.

- Next, in Windows Active Directory, enable the 'Create All Child Objects' privilege for the VCS Helper service user account (HAD Helper) on the MSMQ virtual server.

Note: You do not need to add this privilege if the VCS Helper service user account belongs to the Domain Administrator group.

- Keep the following information ready; the wizard will prompt you for these details:
 - A unique virtual server name for the MSMQ server.
 - A unique virtual IP address for the MSMQ server.
 The virtual IP address is required only if you wish to configure an IPv4 address. In case of IPv6, the wizard prompts you to select the IPv6 prefix and automatically generates an IPv6 address that is valid and unique on the network. The wizard uses the prefix that is advertised by the router on the IPv6 network.

Note: Ensure that there is only one IP resource per MSMQ resource. If there are multiple MSMQ resources that use the same IP resource, only one MSMQ resource will remain online, and the rest will go into the unknown state.

See [“Binding an MSMQ instance to the correct IP address”](#) on page 214.

Configuring the MSMQ resource using the command-line utility

Complete the following steps to configure an MSMQ service group using the MSMQ configuration utility. Make sure that you review the resource types and attribute definitions of the MSMQ agent in the *Cluster Server Bundled Agents Reference Guide*.

To configure an MSMQ resource using the command-line utility

- 1 Start VCS on all systems.
- 2 Ensure that all the required resources are online.
- 3 Ensure that the volume or LUN for the MSMQ data is mounted or connected on the node on which you are configuring the MSMQ resource.

4 Run the MSMQ configuration utility for VCS.

At the command prompt type:

```
msmqconfig -c -n MSMQResourceName -s nodesInServiceGroupSystemList
-m storagePath
```

Here, *storagePath* is the volume on which the MSMQ data is stored. The MSMQ storage path must be created before you run this utility.

For example, if the MSMQ resource name is **vxmsmq**, nodes in the system list are **S1** and **S2**, and the storage path created is **X:\MSMQ\Storage**, then the command will be as follows:

```
msmqconfig -c -n vxmsmq -s S1 S2 -m X:\MSMQ\Storage
```

This will accomplish the preconfiguration, and create the registry key **HKLM\Software\Microsoft\MSMQ\Clustered QMs\MSMQ\$vxmsmq** for RegRep.

The MSMQ configuration utility stops and restarts the default MSMQ service.

5 From the Java console, add a MountV resource (in case of SFW HA) or a NetAppSnapDrive resource (in case of VCS) for **x:**.

A VMDg resource (in case of SFW HA) or a NetAppFiler resource (in case of VCS) may be required if the existing VMDg or NetAppFiler resources do not have a spare volume or LUN to host **x:**.

6 From the Java console add a RegRep resource with the registry key created in step 4 earlier.

Add a MountV and a VMDg resource (in case of SFW HA) or a NetAppSnapDrive and a NetAppFiler resource (in case of VCS), if they do not exist already, for storing the registry replication information required by RegRep.

7 From the Java console, add an MSMQ resource and set the following:

- IPResName to the existing IP resource name
- LanmanResName to the existing Lanman resource name
- StoragePath to the storage path for MSMQ (Referring to the example in step 3, the storage path will be **\\MSMQ\\Storage**.)
- MountResName to the MountV resource (in case of SFW HA) or FilerResName to the NetAppSnapDrive resource (in case of VCS) that you added in step 4.

You must disable and enable the MSMQ resource every time you make changes to the IPResName attribute.

8 Bring the MSMQ service group online.

- 9** Run the virtual MMC Viewer (`VCSVMCView.exe`) on the node where the MSMQ resource is online.

At the command prompt, type:

```
VCSVMCView.exe -target MSMQ
```

You will be prompted for the VirtualName of the LanmanResName provided in the MSMQ resource.

The virtual MMC viewer enables you to create and modify message queues from the node on which the MSMQ resource is online.

- 10** Click **Yes** to launch the virtual server MMC.

You can now create, delete, and modify message queues on the virtual MSMQ.

Configuring the MSMQ service group using the wizard

Complete the following steps to configure an MSMQ service group using the MSMQ Configuration Wizard. Make sure you review the resource types and attribute definitions of the MSMQ agent in the *Cluster Server Bundled Agents Reference Guide*.

To configure an MSMQ service group using the MSMQ Configuration Wizard

- 1** Start the MSMQ Configuration Wizard.

Click **Start > All Programs > Veritas > Veritas Cluster Server > Configuration Tools > MSMQ Configuration Wizard**.

On Windows 2012 operating systems, use the **Apps** menu.
- 2** Review the information on the Welcome panel and then click **Next**.
- 3** On the Wizard Options panel click **Create service group** and then click **Next**.

- 4 On the Service Group Configuration panel, specify the service group name, choose the systems for the service group, and then click **Next**.

Specify the following details:

Service Group Name Type a name for the MSMQ service group.

Available Cluster Systems Select the systems on which to configure the service group and click the right arrow to move the systems to the service group's system list.

To remove a system from the service group's system list, click the system in the Systems in Priority Order box and click the left arrow.

To change a system's priority in the service group's system list, click the system from the Systems in Priority Order and click the up and down arrows.

System priority defines the order in which service groups are failed over to systems. The system at the top of the list has the highest priority while the system at the bottom of the list has the lowest priority.

Include selected systems in the service group's AutoStartList attribute To enable the service group to automatically come online on one of the systems, select this checkbox.
 See ["Service group attributes"](#) on page 546.

- 5 On the Virtual Server Configuration panel, specify information related to your network and then click **Next**.

Do the following:

- Select **IPv4** to configure an IPv4 address for the MSMQ virtual server.
 - In the Virtual IP Address field, type a unique virtual IPv4 address for the MSMQ virtual server.
 - In the Subnet Mask field, type the subnet to which the virtual IPv4 address belongs.
- Select **IPv6** to configure an IPv6 address for the virtual server. The IPv6 option is disabled if the network does not support IPv6.
 - Select the prefix from the drop-down list. The wizard uses the prefix and automatically generates an IPv6 address that is valid and unique on the network.

- In the Virtual Server name field, type a unique name for the MSMQ virtual server. This is the name by which clients will connect to the MSMQ server. The virtual name must not exceed 15 characters.
- For each system in the cluster, select the public network adapter name. This field displays the TCP/IP enabled adapters on a system, including the private network adapters, if applicable. To view the adapters associated with a system, click the Adapter Display Name field and click the arrow. Verify that you select the adapters assigned to the public network, not the private.
- Click **Advanced Settings** to specify additional details for the Lanman resource.

On the Lanman Advanced Configuration dialog box, do the following:

- Check **Active Directory Update required** check box to enable the Lanman resource to update the Active Directory with the virtual name. This sets the Lanman agent attributes ADUpdateRequired and ADCriticalForOnline to true.
- In the Organizational Unit field, type the distinguished name of the Organizational Unit for the virtual server in the format
 CN=containername,DC=domainname,DC=com.
 To browse for an OU, click ... (ellipsis button) and search for the OU using the Windows Find Organization Units dialog box.
 By default, the Lanman resource adds the virtual server to the default container "Computers."
- Click **OK**.
 The user account for VCS Helper service must have adequate privileges on the specified container to create and update computer accounts.

- 6 On the MSMQ and RegRep Directory Details panel, specify the MSMQ and registry replication directories and then click **Next**.

Specify the following details:

MSMQ Directory

Specify the directory path for storing the MSMQ data. You can either type the path or click ... (ellipsis button) to browse for a directory.

The MSMQ agent uses the specified MSMQ directory to store all the message queues.

Replication Directory

Specify the directory path for storing the MSMQ registry data. You can either type the path or click ... (ellipsis button) to browse for a directory.

The Registry Replication agent uses the specified regrep directory to store the MSMQ registry related information.

Note: Make sure that these are new or empty directories. They must not contain data pertaining to an MSMQ service group that was deleted.

- 7 This is applicable in case of VCS for Windows and in a NetApp storage environment.

On the Initiator Selection panel, select the initiator for the virtual disk from the list of available initiators displayed for each cluster node, and then click **Next**.

If you are configuring multipath I/O (MPIO) over Fibre Channel (FC), you must select at least two FC initiators for each cluster node. Note that the node from which you run this wizard already has an initiator selected by default. This is the initiator that was specified when you connected the LUNs to this cluster node.

- 8 On the Service Group Summary panel, review the service group configuration and click **Next**.

The following service group details are visible:

Resources	<p>Displays a list of configured resources. The wizard assigns unique names to resources. Change the names of resource, if required.</p> <p>To edit a resource name, select the resource name and either click it or press the F2 key. Edit the resource name and then press the Enter key to confirm the changes. To cancel editing a resource name, press the Esc key.</p>
Attributes	<p>Displays the attributes and their configured values, for a resource selected in the Resources list.</p>

- 9 Click **Yes** on the dialog that prompts you that the wizard will run commands to modify the service group configuration.
- 10 In the completion dialog box, check **Bring the service group online** check box if you want to bring the service group online on the local system, and then click **Finish**.

This completes the MSMQ service group configuration.

You can now create, delete, and modify message queues on the virtual MSMQ. Use the VCS Application Manager utility.

See [“About the VCS Application Manager utility”](#) on page 241.

Modifying an MSMQ service group using the wizard

The MSMQ Configuration Wizard enables you to modify an MSMQ service group. Consider the following before you modify MSMQ service groups using the wizard:

- If the MSMQ service group is online, you must run the wizard from a system on which the service group is online. You can then add and remove resources to the configuration using the wizard; you cannot modify resources that are online.
- To change the online resource attributes, you must take the service group offline. However, the MountV and VMDg (in case of SFW HA), Mount and DiskRes (in case of Windows LDM) and NetAppSnapDrive and NetAppFiler (in case of VCS for Windows) resources for the service group should be online on the node where you run the wizard and offline on all other nodes.

- If you are running the wizard to remove a node from the service group's system list, do not run the wizard on the node being removed.
- If the service group contains resources that were not part of the default service group configuration, then modifying the service group may change those resources. You may then have to manually restore the settings of those resources later.
- If you are modifying the service group to remove an MSMQ resource, make sure you offline the resource before deleting it.

To modify an MSMQ service group using the MSMQ Configuration Wizard

- 1** Start the MSMQ Configuration Wizard.
 Click **Start > All Programs > Veritas > Veritas Cluster Server > Configuration Tools > MSMQ Configuration Wizard**.
 On Windows 2012 operating systems, use the **Apps** menu.
- 2** Read the information on the Welcome panel and click **Next**.
- 3** On the Wizard Options panel, click **Modify service group**, select the service group to be modified, and click **Next**.
- 4** Follow the wizard instructions and make desired modifications to the service group configuration.
 See ["Configuring the MSMQ service group using the wizard"](#) on page 208.

Configuring MSMQ agent to check port bindings more than once

The MSMQ agent verifies that the clustered MSMQ service is bound to the correct virtual IP and port. By default, the agent performs this check only once during the Online operation. If the clustered MSMQ service is not bound to the correct virtual IP and port, the agent stops the service and the resource faults. You can configure the number of times that this check is performed.

To configure MSMQ agent to check port bindings more than once

- 1** Open the registry editor and navigate to the following location:
`HKEY_LOCAL_MACHINE\SOFTWARE\VERITAS\VCS\BundledAgents\MSMQ`
 If the `BundledAgents` and `MSMQ` keys are not present, create them.
- 2** Add a tunable parameter named `VirtualIPPortCheckRetryCount`.

- 3 Set the `DWORD` value of this parameter to a number greater than 1.
- 4 Save and exit the registry editor.

The agent starts the clustered MSMQ service again and verifies its virtual IP and port binding as many times as defined in the `VirtualIPPortCheckRetryCount` parameter. It waits 2 seconds between each verification attempt. If the clustered MSMQ service is bound to the correct virtual IP and port, the agent reports Online.

Binding an MSMQ instance to the correct IP address

A clustered MSMQ resource fails to come online on a Windows Server 2012 or 2012 R2 system, because the default MSMQ service does not bind to the correct IP address and port.

Check whether the default MSMQ service has the correct network bindings by running the `netstat` command as follows:

```
netstat -abno | findstr 1801
```

If the network bindings are incorrect, the output of this command appears similar to that in the following sample:

```
TCP    0.0.0.0:1801    0.0.0.0:0      LISTENING      4148
TCP    [::]:1801      [::]:0         LISTENING      4148
```

To bind an MSMQ instance to the correct IP address

- 1 Create the `BindInterfaceIP` registry key for the default MSMQ service by following these steps sequentially:
 - Open the registry.
 - If Directory Service Integration is enabled for MSMQ, go to `HKEY_LOCAL_MACHINE\Software\Microsoft\MSMQ\Parameters`.
 Create a new String value named `BindInterfaceIP`.
 For the Value data, enter the system IP address.
 - If Directory Service Integration is *not* enabled, go to `HKEY_LOCAL_MACHINE\Software\Microsoft\MSMQ\ClusteredQMs\MSMQ$resourceName\Parameters`.
 Replace the `resourceName` variable with the appropriate MSMQ resource name.
 Create a new String value named `BindInterfaceIP`.
 For the Value data, enter the virtual IP address.

Note: If you do not specify the virtual IP address, the MSMQ service fails to come online.

For more information, see the Microsoft Knowledge Base [Article 974813](#).

Note: For a Windows Server 2012 or 2012 R2 system, you do not need to install the hotfix mentioned in the article.

- 2 Restart the default MSMQ service.
- 3 Check whether the default MSMQ service binds to `systemIP:1801`.
See “[Checking whether MSMQ is listening for messages](#)” on page 215.
- 4 Bring the clustered MSMQ resource or the service group online.
- 5 Check whether the clustered MSMQ service has the correct network bindings.
See “[Checking whether MSMQ is listening for messages](#)” on page 215.

Checking whether MSMQ is listening for messages

If the MSMQ driver is not listening on port 1801, it will not receive messages. In particular, the clustered MSMQ driver (the virtual instance) must be listening for message flow that may be necessary between clustered MSMQ resources. To check whether the clustered instance of the MSMQ service is listening, perform this procedure on the cluster nodes.

To check whether MSMQ is listening

- 1 Run the `netstat` command as follows:

```
netstat -abno | findstr 1801
```

All instances of the MSMQ service that are bound to local IP addresses are listed, which includes the local MSMQ service and the clustered MSMQ instances for various applications.

A sample output is as follows:

TCP	10.251.59.54:1801	0.0.0.0:0	LISTENING	4148
TCP	10.251.59.118:1801	0.0.0.0:0	LISTENING	102556
TCP	10.251.59.117:1801	0.0.0.0:0	LISTENING	75272

The three different process IDs in this sample indicate that three different message queue drivers are running.

- 2 To confirm that one of these is the virtual driver in use for the clustered application, run the `tasklist` command as follows:

```
tasklist /svc | findstr processID
```

For example:

```
tasklist /svc | findstr 75272
```

A sample output is as follows:

```
mqsvc.exe          75272  MSMQ$AppSG-MSMQ
```

This indicates that the virtual message queue driver has bound to the port where MSMQ is expected to run. MSMQ\$AppSG-MSMQ is the name of the clustered driver in the application service group. If this is not present, the application will not be able to send and receive messages.

About configuring the infrastructure and support agents

Following is an overview of the steps to configure the VCS infrastructure and support agents:

On Windows Server Core systems, you have to add the required resources and configure the service group manually. You can perform the steps either directly on the Server Core machine using the VCS commands, or remotely using the Cluster Manager (Java console).

See [“About administering VCS from the command line”](#) on page 97.

Note: If you have configured a firewall, add ports 14141 and 14150 to the exceptions list.

Before configuring the service group, review the resource types and the attribute definitions of the agents, described in the *Cluster Server Bundled Agents Reference Guide*.

About configuring notification

Use the NotifierMngr agent to set up notification in your cluster. Review the information about how VCS handles notification.

See [“About VCS event notification”](#) on page 357.

VCS provides a wizard to set up notification.

Configuring registry replication

The Registry Replication (RegRep) agent replicates the registry of the active cluster node.

To configure registry replication

- 1 Configure an exclusive MountV resource (in case of SFW HA), or a Mount resource (in case of Windows LDM), or a NetAppSnapDrive resource (in case of VCS for Windows) for the Registry Replication agent. Verify that no other applications use this resource.

See [“About storage configuration”](#) on page 161.

- 2 Create a resource of type RegRep.

- 3 Configure the following required attributes for the RegRep resource.

- **Keys:** The list of registry keys to be monitored. From the 'name-value' pair of a registry key, you must provide the name of the registry keys to be synchronized and not the value for that key.

When defining the keys, you must use abbreviations.

See [“About registry hive abbreviations”](#) on page 218.

Instructions on how to exclude certain keys from being replicated are available.

See [“About excluding keys”](#) on page 218.

Instructions on how to replicate registry keys without replicating the subkey are available.

See [“About ignoring subkeys”](#) on page 219.

Do not configure more than 63 keys for a single RegRep resource otherwise the resource will go in an unknown state.

- **MountResName or FilerResName:** The name of the MountV resource (in case of SFW HA) or Mount resources (in case of Windows LDM), or NetAppSnapDrive resource (in case of VCS for Windows) on which the Registry Replication agent depends. The resource specifies the mount drive or LUN on the shared disk where the log file is created.
 - **ReplicationDirectory:** The directory on the shared disk in which the registry changes are logged.
- 4 Configure other resources for the service group, if required.
 - 5 Link the RegRep and MountV (in case of SFW HA), or Mount (in case of Windows LDM), or NetAppSnapDrive (in case of VCS for Windows) resources such that the RegRep resource depends on the MountV, Mount, or NetAppSnapDrive resource.
 - 6 Bring the RegRep resource, and other resources in the service group, online.

About registry hive abbreviations

To configure a registry key to be replicated or excluded, use the abbreviation corresponding to the registry hive as described in the following table.

[Table 7-1](#) shows the abbreviation corresponding to the registry hive.

Table 7-1 RegRep agent - Registry hive and abbreviations

Registry Hive	Abbreviation to use
HKEY_LOCAL_MACHINE	HKLM
HKEY_CURRENT_USER	HKCU
HKEY_USERS	HKU
HKEY_CURRENT_CONFIG	HKCC
HKEY_CLASSES_ROOT	HKCR

About excluding keys

This topic describes the algorithm the Registry Replication agent uses while excluding keys. For example, assume a registry key KEY_X has a subkey of KEY_Y, which has another subkey KEY_Z. This key would appear as KEY_X\KEY_Y\KEY_Z in the Registry Editor. The following table describes various scenarios of keys

marked for replication and for exclusion. The Result column describes the agent behavior in these scenarios.

[Table 7-2](#) shows Registry Replication exclude keys and behavior.

Table 7-2 RegRep agent - Exclude keys and behavior

Keys for replication	Exclude keys	Result
KEY_X	KEY_Y\KEY_Z	KEY_Y is excluded, so is KEY_Z.
KEY_X	KEY_Y	KEY_Y is excluded, so is KEY_Z.
KEY_X	KEY_X	KEY_X is not excluded and an error message is logged.
KEY_X\KEY_Y	KEY_X	KEY_X is not excluded and an error message is logged.

About ignoring subkeys

Use the IgnoreSubKeys option for the Keys attribute to prevent the Registry Replication agent from replicating the subkeys. The following table describes possible combination of values for the Keys attribute. The Result column describes the agent behavior in these scenarios:

[Table 7-3](#) shows the IgnoreSubKeys and their behavior for the Registry replication agent.

Table 7-3 RegRep agent - IgnoreSubKeys and behavior

Value specified for "Keys" attribute	Result
"HKLM\SOFTWARE\VERITAS\VCS"	Replicates the subkeys
"HKLM\SOFTWARE\VERITAS\VCS"=IgnoreSubKeys	Does not replicate the subkeys
"HKLM\SOFTWARE\VERITAS\VCS"=IgnoreSubKeys:Yes	Does not replicate the subkeys
"HKLM\SOFTWARE\VERITAS\VCS"=IgnoreSubKeys:No	Replicates the subkeys
"HKLM\SOFTWARE\VERITAS\VCS"=<any other value>	Replicates the subkeys

About additional considerations for using IgnoreSubKeys

Veritas recommends not to set the IgnoreSubKeys value when the RegRep resource is online. Even if the value is set with the resource online, the changes will be applicable after the next online function.

Configuring a proxy resource

The Proxy agent monitors and mirrors the state of a resource on a local or remote system in a VCS cluster. Use this agent to reduce overheads in configurations where multiple resources point at the same physical device. For example, if multiple service groups use the same NIC, configure one service group to monitor the NIC and have Proxy resources in the other service groups to mirror the state of the NIC resource.

To configure a proxy resource

- 1 Create a resource of type Proxy.
- 2 Configure the following required attribute for the Proxy resource:
 - TargetResName: The name of the target resource whose status is to be monitored and mirrored by the Proxy resource.If required, configure the following optional attribute for the Proxy resource:
 - TargetSysName: The name of the system associated with the target resource. If this attribute is not specified, the Proxy resource assumes the system is local.
- 3 Configure other resources for the service group, if required.
- 4 Bring the Proxy resource, and other resources in the service group, online.

Configuring a phantom resource

A Phantom resource enables VCS to determine the status of service groups that do not include OnOff resources.

To configure a phantom resource

- 1 Create a resource of type Phantom.
- 2 Configure other resources for the service group, if required.
- 3 Bring the Phantom resource, and other resources in the service group, online.

Configuring file resources

The FileNone, ElifNone, FileOnOff, and FileOnOnly agents help you test VCS functionality as follows:

- The FileNone agent monitors a file and returns ONLINE if the file exists.
 - The ElifNone agent monitors a file and returns ONLINE if the file does not exist.
 - The FileOnOff agent creates, removes, and monitors a file.
 - The FileOnOnly agent creates and monitors a file.
- The process of configuring these resources is similar.

To configure a file resource

- 1 In your service group, create a resource of the desired type.
- 2 Configure the required attribute PathName for the resource.
- 3 If required, configure additional resources in the service group.
- 4 Bring the file resource, and other resources, in the service group online.

Configuring a RemoteGroup resource

The RemoteGroup agent establishes dependencies between applications that are configured on different VCS clusters. With the RemoteGroup agent you can monitor or manage a service group that exists in a remote cluster.

Some points about configuring the RemoteGroup resource are as follows:

- For each remote service group that you want to monitor or manage, you must configure a corresponding RemoteGroup resource in the local cluster.
- Multiple RemoteGroup resources in a local cluster can manage corresponding multiple remote service groups in different remote clusters.
- You can include the RemoteGroup resource in any kind of resource or service group dependency tree.
- A combination of the state of the local service group and the state of the remote service group determines the state of the RemoteGroup resource.

Before configuring the RemoteGroup resource, review the resource types, the attribute definitions, and the sample scenario described in the *Cluster Server Bundled Agents Reference Guide*.

To configure a RemoteGroup resource

- 1 In your service group, create resources of type IP and NIC.
- 2 Create a resource of type RemoteGroup.

- 3 Configure the required attributes for the RemoteGroup resource. See the *Cluster Server Bundled Agents Reference Guide* for more information on the required attributes and their definitions.
- 4 Link the resources as follows:
 - Link the IP and NIC resources such that the IP resource depends on the the NIC resource.
 - Link the RemoteGroup and NIC resources such that the RemoteGroup resource depends on the NIC resource.
- 5 Configure other resources in the service group, if required.
- 6 Bring the IP, NIC, and RemoteGroup resources online.

About configuring applications using the Application Configuration Wizard

VCS provides an Application Configuration Wizard to create service groups to monitor applications that are configured as resources of type GenericService, ServiceMonitor, or Process. You can also use the wizard to add registry replication and network resources to application service groups.

Note: The wizard does not configure the registry replication and network resources independently. It configures these resources as part of a service group that has application resources.

On Windows Server Core, you have to add the required resources and configure the service group manually. You can perform the steps either directly on the Server Core machine using the VCS commands, or remotely using the Cluster Manager (Java console).

See [“About administering VCS from the command line”](#) on page 97.

Before configuring the service group, review the resource types and the attribute definitions of the agents, described in the *Cluster Server Bundled Agents Reference Guide*.

Before you configure service groups using the Application Configuration wizard

Note the following prerequisites before you configure application service groups using the Application Configuration wizard:

- Verify that the application you wish to configure is installed on the nodes that are going to be part of the service group.
- Verify that the startup type of the application service that you wish to configure is set to manual on all the nodes that are going to be part of the service group.
- Verify that the application service is stopped on all the nodes that are going to be part of the service group.
- Verify that the shared drives or LUNs required by the applications are mounted on the node where you run the wizard.
 See [“About managing storage using Windows Logical Disk Manager”](#) on page 162.
 See [“About managing storage in a Network Appliance storage environment”](#) on page 166.
 See [“About managing shared storage using Storage Foundation for Windows”](#) on page 168.
- If you have configured a firewall, add the required ports and services to the Firewall Exception list.
 For a detailed list of services and ports used, refer to the product installation and upgrade guide.
- Before running the wizard, make sure you have the following information ready:
 - Details of the application that you wish to configure (for example, application type, service name, start parameters, startup directory)
 - Shared storage used by the applications
 - Application registry entries for configuring registry replication
 - Network and virtual computer (Lanman) details for the application

Note: These prerequisites apply to Application Configuration Wizard. For agent-specific prerequisites, see the agent descriptions in the *Cluster Server Bundled Agents Reference Guide*.

Adding resources to a service group

This topic describes how to use the Application Configuration Wizard to add resources to a service group.

To add resources to a service group

- 1 Start the Application Configuration Wizard.
 On Windows 2012 operating systems, use the **Apps** menu.
- 2 Review the information on the Welcome panel and click **Next**.

- 3 On the Wizard Options panel, click **Create service group** and click **Next**.
- 4 On the Service Group Configuration panel, specify the following service group details and then click **Next**:

Service Group Name Type a name for the service group.

Available Cluster Systems Select the systems on which to configure the service group and click the right arrow to move the systems to the service group's system list.

To remove a system from the service group's system list, click the system in the Systems in Priority Order box and click the left arrow.

To change a system's priority in the service group's system list, click the system from the Systems in Priority Order and click the up and down arrows.

System priority defines the order in which service groups are failed over to systems. The system at the top of the list has the highest priority while the system at the bottom of the list has the lowest priority.

Include selected systems in the service group's AutoStartList attribute To enable the service group to automatically come online on one of the systems, select this checkbox.
 See ["Service group attributes"](#) on page 546.

- 5 The Application Options dialog box provides you the option to specify the type of application to be configured.

The following options are available:

Generic Service Configures a service using the Generic Service agent. The agent brings services online, takes them offline, and monitors their status.
 See ["Configuring a GenericService resource"](#) on page 225.

Process Configures a process using the Process agent. The agent brings processes online, takes them offline, and monitors their status.
 See ["Configuring processes"](#) on page 226.

Service Monitor Configures a service using the ServiceMonitor agent. The agent monitors a service or starts a user-defined script and interprets the exit code of the script.
 See ["Adding resources to a service group"](#) on page 223.

Configuring a GenericService resource

This topic describes how to use the Application Configuration Wizard to configure a GenericService resource.

To configure a GenericService resource

- 1 In the Application Options panel, click **Create**, select **GenericService** from the corresponding drop-down list, and click **Next**.
- 2 On the Generic Service Options panel, specify the details of the service that you wish to configure and then click **Next**.

Specify the service for which you wish to configure a GenericService resource and then specify the following attributes:

- Click the ... (ellipsis button) adjacent to the Service Name text box.
- In the Services dialog box, select a service and click **OK**. The selected service appears in the Service Name text box.
- In the Start Parameters text box, provide the start parameters for the service, if any.
- In the Delay After Online text box, specify the number of seconds the agent waits after the service is brought online before starting the monitor function.
- In the Delay After Offline text box, specify the number of seconds the agent waits after the service is taken offline before starting the monitor function.

- 3 On the User Details panel, specify the details of the user in whose context the service will run and then click **Next**.

Do the following:

- To configure a service to run in the context of a local system account, click **Local System account**.
- To configure a service to run in the context of another user account, click **This Account** and then specify the **Domain Name**, **User Name**, and **Password** in the respective fields.

- 4 On the Shared Storage Option panel, under Available Shared Drives box, select the check box adjacent to the shared drive and then click **Next**.

This is the shared storage that is required by the GenericService resource. The shared storage that you select will be in addition to the mount where the service binaries exist.

- 5 In the Application Resource Summary panel, review the summary of the GenericService resource. Click **Back** to make changes. Otherwise, click **Next**.
- 6 In the Application Options dialog box, select one of the following options:

- To configure another GenericService resource, repeat step [To configure a GenericService resource](#) through step [To configure a GenericService resource](#).
- To configure a Process resource:
See [“Configuring processes”](#) on page 226.
- To configure a ServiceMonitor resource:
See [“Configuring a ServiceMonitor resource”](#) on page 228.
- To configure other resources, including FileShare, Registry Replication, and Network resources:
See [“Configuring VCS components”](#) on page 229.

If you do not wish to add any more resources, proceed to configuring the service group.

See [“Configuring service groups using the Application Configuration Wizard”](#) on page 232.

Configuring processes

This topic describes how to use the Application Configuration Wizard to configure processes.

To configure processes

- 1 In the Application Options panel, click **Create**, select **Process** from the corresponding list, and click **Next**.
- 2 On the Process Details panel, specify the details of the process that you wish to configure and then click **Next**.

Specify the process details as follows:

- In the Start Program text box, specify the complete path of the program that will start the process to be monitored by VCS. You can choose to either type the location of the program or browse for it using ... (ellipsis button).
- In the Start Program Parameters text box, specify the parameters used by the Process agent start program.
- In the Program Startup Directory text box, type the complete path of the Process agent program or browse for it by clicking ... (ellipsis button).
- In the Stop Program text box, type the complete path of the program that will stop the process started by the Start Program or browse for it by clicking ... (ellipsis button).
- In the Stop Program Parameters text box, specify the parameters used by the stop program.

- In the Monitor Program text box, type the complete path of the program that monitors the Start Program or browse for it by clicking ... (ellipsis button). If you do not specify a value for this attribute, VCS monitors the Start Program. If the Start Program is a script to launch another program, you must specify a monitor program.
 - In the Monitor Program Parameters text box, specify the parameters used by the monitor program.
 - In the Clean Program text box, type the complete path of the Clean process or browse for it by clicking ... (ellipsis button). If no value is specified, the agent kills the process indicated by the Start Program.
 - In the Clean Program Parameters text box, specify the parameters used by the Clean program.
 - Check the **Process interacts with the desktop** check box if you want the process to interact with your Windows desktop. Setting this option enables user intervention for the process.
- 3** On the User Details panel, specify information about the user in whose context the process will run and then click **Next**.
- Do the following:
- To configure a service to run in the context of a local system account, click **Local System account**.
 - To configure a service to run in the context of another user account, click **This Account** and then specify the **Domain Name**, **User Name**, and **Password** in the respective fields.
 - Click **Next**.
- 4** On the Shared Storage Option panel, under Available Shared Drives box, select the check box adjacent to the shared drive and then click **Next**.
- This is the shared storage required by the Process resource. The shared storage that you select will be in addition to the mount where the process binaries exist.
- 5** In the Application Resource Summary panel, review the summary of the Process resource. Click **Back** to make changes. Otherwise, click **Next**.
- 6** In the Application Options dialog box, select one of the following options:
- To configure another Process resource, repeat step **1** through step **5**.
 - To configure a GenericService resource:
See "[Configuring a GenericService resource](#)" on page 225.

- To configure a ServiceMonitor resource:
 See [“Configuring a ServiceMonitor resource”](#) on page 228.
- To configure other resources, including Registry Replication and Network resources:
 See [“Configuring VCS components”](#) on page 229.
 If you do not want to add any more resources, proceed to configuring the service group.
 See [“Configuring service groups using the Application Configuration Wizard”](#) on page 232.

Configuring a ServiceMonitor resource

This topic describes how to use the Application Configuration Wizard to configure a ServiceMonitor resource.

To configure a ServiceMonitor resource

- 1 In the Application Options panel, click **Create**, select **ServiceMonitor** from the corresponding drop-down list, and click **Next**.
- 2 Specify the service to be monitored or a user-defined script to monitor a service.
 If you want VCS to monitor the service, do the following:
 - Select the **Service** option and click ... (ellipsis button) adjacent to the Service Name text box.
 - In the Service dialog box, select the service and click **OK**. The selected service name appears in the Service Name text box. Alternatively, you may also type the service name to be monitored.
 - Click **Next**.

If you want a script to monitor the service, do the following:

- Click ... (ellipsis button) and specify the complete path for the script.
 - Specify the parameters for the script.
 - Specify the time in seconds for the agent to receive a return value from the monitor script.
 - Click **Next**.
- 3 On the User Details panel, specify the user information in whose context the service will be monitored.
 Do the following:
 - To configure a service to run in the context of a local system account, click **Local System account**.

- To configure a service to run in the context of another user account, click **This Account** and then specify the **Domain Name**, **User Name**, and **Password** for the user account.
If the service selected in step 2 is running in the context of a local system account, the **This Account** option is disabled. Similarly, if the service is running in the context of any other user account, the **Local System account** option is disabled.
 - Click **Next**.
Service Monitor resource belongs to the category of persistence resources. Such resources do not depend on other VCS resources, including shared storage. Hence, the Shared Storage Option dialog box does not appear if you select the ServiceMonitor option.
- 4** In the Application Resource Summary panel, review the summary of the ServiceMonitor resource. Click **Back** to make changes. Otherwise, click **Next**.
- 5** In the Application Options dialog box, select one of the following options:
- To configure another ServiceMonitor resource, repeat step 1 through step 4.
 - To configure a GenericService resource:
See [“Configuring a GenericService resource”](#) on page 225.
 - To configure a Process resource:
See [“Configuring processes”](#) on page 226.
 - To configure other resources, including Registry Replication and Network resources:
See [“Configuring VCS components”](#) on page 229.
If you do not want to add any more resources, proceed to configuring the service group.
See [“Configuring service groups using the Application Configuration Wizard”](#) on page 232.

Configuring VCS components

Applications configured using GenericService or Process resources may require network components or registry replication resources. You can configure these VCS components only for service groups created using the wizard.

Note: Configure these components only after configuring all application resources. The wizard creates a service group after these components are configured. To add more application resources, you must rerun the wizard in the Modify mode.

To configure VCS components

1 In the Application Options panel, click **Configure Other Components**.

2 Select the VCS component to be configured for your applications.

The available options are as follows:

- **Registry Replication Component:** Select this option to configure registry replication for your application. To configure a Registry Replication resource, proceed to step [3](#).
- **Network Component:** Select this option to configure network components for your application. If you wish to configure a virtual computer name, check **Lanman component** also. To configure a network resource, proceed to step [5](#).

The wizard does not enable the **Lanman Component** check box unless the **Network Component** check box is checked.

3 Specify the registry keys to be replicated.

The RegistryReplication dialog box appears only if you chose to configure the Registry Replication Component in the Application Component dialog box.

- Specify the directory on the shared disk in which the registry changes are logged.
- Click **Add**.
- In the Registry Keys dialog box, select the registry key to be replicated.
- Click **OK**. The selected registry key is added to Registry KeyList box.
- This is applicable in case of VCS for Windows only.
Check the **Configure NetApp SnapMirror Resource(s)** check box if you want to set up a disaster recovery configuration. The SnapMirror resource is used to monitor replication between filers at the primary and the secondary site, in a disaster recovery configuration. Note that you must configure the SnapMirror resource only after you have configured the cluster at the secondary site.
- Click **Next**.

If you chose Network Component from the Application Component dialog box, proceed to the next step. Otherwise, proceed to step [6](#).

- 4** This step is applicable in case of VCS for Windows only.

On the Initiator Selection panel, select the initiator for the virtual disk from the list of available initiators displayed for each cluster node, and then click **Next**.

If you are configuring multipath I/O (MPIO) over Fibre Channel (FC), you must select at least two FC initiators for each cluster node. Note that the node from which you run this wizard already has an initiator selected by default. This is the initiator that was specified when you connected the LUNs to this cluster node.

- 5** The Virtual Computer Configuration dialog box appears only if you chose to configure the Network Component in the Application Component dialog box. Specify the network related information as follows:

- Select **IPv4** to configure an IPv4 address for the virtual server.
 - In the Virtual IP Address field, type a unique virtual IPv4 address for the virtual server.
 - In the Subnet Mask field, type the subnet to which the virtual IPv4 address belongs.
- Select **IPv6** to configure an IPv6 address for the virtual server. The IPv6 option is disabled if the network does not support IPv6.
 - Select the prefix from the drop-down list. The wizard uses the prefix and automatically generates an IPv6 address that is valid and unique on the network.
- In the Virtual Server Name field, enter a unique virtual computer name by which the node will be visible to the other nodes.
The virtual name must not exceed 15 characters. Note that the Virtual Computer Name text box is displayed only if you chose to configure the Lanman Component in Application Component dialog box.
- For each system in the cluster, select the public network adapter name. To view the adapters associated with a system, click the **Adapter Display Name** field and click the arrow.
Note that the wizard displays all TCP/IP enabled adapters on a system, including the private network adapters, if applicable. Ensure that you select the adapters assigned to the public network, not the private.
- Click **Advanced** and then specify additional details for the Lanman resource as follows:
 - Check **AD Update required** to enable the Lanman resource to update the Active Directory with the virtual name.

This sets the Lanman agent attributes ADUpdateRequired and ADCriticalForOnline to true.

- In the Organizational Unit field, type the distinguished name of the Organizational Unit for the virtual server in the format
`CN=containername,DC=domainname,DC=com.`
 To browse for an OU, click ... (ellipsis button) and search for the OU using the Windows Find Organization Units dialog box. By default, the Lanman resource adds the virtual server to the default container "Computers."
 The user account for VCS Helper service must have adequate privileges on the specified container to create and update computer accounts.
- Click **OK**.
- Click **Next**.

6 In the Application Options dialog box, select one of the following options:

- To configure additional VCS components, repeat step 1 through step 5.
- To configure a GenericService resource:
 See ["Configuring a GenericService resource"](#) on page 225.
- To configure a Process resource:
 See ["Configuring processes"](#) on page 226.
- To configure a Service Monitor resource:
 See ["Configuring a ServiceMonitor resource"](#) on page 228.

If you do not want to add any more resources, proceed to configuring the service group:

See ["Configuring service groups using the Application Configuration Wizard"](#) on page 232.

Configuring service groups using the Application Configuration Wizard

The Application Configuration Wizard enables you to create service group for the application resources and other VCS components configured using the wizard. This topic describes how to create the service group using the wizard.

To configure a service group using the wizard

- 1** In the Application Options panel, click **Configure application dependency and create service group**.

The option is enabled only if the following conditions are met:

- Resources and VCS components are already configured using the wizard.

- You clicked **Modify Service Groups** in the Wizard Options panel.

2 Specify the dependency between the applications.

You must have at least two resources configured for this dialog box to appear. Of the two resources, one should either be a GenericService or a Process resource.

- From the Select Application list, select the application that would depend on other applications. The selected application becomes the parent application.
- From the Available Applications list, select the application on which the parent application would depend and click the right-arrow icon to move the application to the Child Applications list.
To remove an application from the Child Applications list, select the application in the list and click the left arrow.
- Repeat these steps for all such applications for which you want to create a dependency.

Click **Next**.

The Application Dependency dialog box enables you to link resources configured using the wizard. If these resources are dependent on other services outside the VCS environment, you should first configure resources for such services and then create the appropriate dependency.

3 On the Service Group Summary panel, review the service group configuration and click **Next**.

The following service group details are visible:

Resources	<p>Displays a list of configured resources. The wizard assigns unique names to resources. Change the names of resource, if required.</p> <p>To edit a resource name, select the resource name and either click it or press the F2 key. Edit the resource name and then press the Enter key to confirm the changes. To cancel editing a resource name, press the Esc key.</p>
Attributes	<p>Displays the attributes and their configured values, for a resource selected in the Resources list.</p>

4 Click **Yes** on the dialog that prompts you that the wizard will run commands to modify the service group configuration.

- 5 In the completion panel, check **Bring the service group online** if you want to bring the service group online on the local system.
- 6 Click **Finish** to create the service group and exit the Application Configuration Wizard.

Modifying an application service group

You can modify an application service group using the Application Configuration Wizard.

Consider the following before you modify service groups using the wizard:

- If the service group to be modified is online, you must run the wizard from a system on which the service group is online. You can then use the wizard to add or remove resources from the configuration. You cannot modify resources that are online.
- To change the resource attributes, you must take the service group offline. However, the MountV and VMDg (in case of SFW HA), Mount and DiskRes (in case of Windows LDM), and NetAppSnapDrive and NetAppFile (in case of VCS for Windows) resources for the service group should be online on the node where you run the wizard and offline on all other nodes.
- If you are running the wizard to remove a node from the service group's system list, do not run the wizard on the node being removed.
- If the service group contains resources that were not part of the default service group configuration, then modifying the service group may change those resources. You may then have to manually restore the settings of those resources later.

Note: Veritas recommends that you do not use the wizard to modify service groups that were not created using the wizard.

To modify an application service group

- 1 Start the Application Configuration Wizard.
 On Windows 2012 operating systems, use the **Apps** menu.
- 2 Review the information on the Welcome panel and click **Next**.
- 3 On the Wizard Options panel, click **Modify service group**. From the Service Groups list, select the service group containing the resource that you want to modify and click **Next**.

- 4 On the Service Group Configuration panel, if required, make changes as appropriate to update the SystemList and AutoStartList attributes, and then click **Next**.

If you want the service group to automatically come online on one of the systems, make sure to select the **Include selected systems in the service group's AutoStartList attribute** checkbox.

- 5 Click **Modify**, select the resource you want to modify and then click **Next**.

The Modify option is enabled only if the following conditions are met:

- Service and Process resources are already configured using the wizard.
- You selected the **Modify Service Groups** option in the Wizard Options panel.

- 6 Depending on the resource you chose to modify from the Application Options page, you would either get the Generic Service Options, Process Details, or the Service Monitor Options dialog box.

Make required changes in the appropriate dialog box and click **Next**.

See [“Configuring a GenericService resource”](#) on page 225.

See [“Configuring processes”](#) on page 226.

See [“Configuring a ServiceMonitor resource”](#) on page 228.

- 7 On the User Details dialog box, specify the user information and click **Next**.
- 8 On the Application Resource Summary dialog box, review the summary of the resource.

See [“About fast failover”](#) on page 157.

Click **Back** to make changes. Otherwise, click **Next**.

- 9 Repeat step 5 through step 8 for each resource that you want to modify.

- 10 After modifying the required resources, you can:

- Add additional resources to the service group.
See [“Adding resources to a service group”](#) on page 223.
- Delete resources from the service group.
See [“Deleting resources from a service group”](#) on page 236.
- Add VCS components to the service group.
See [“Configuring VCS components”](#) on page 229.
- Create the service group.
See [“Configuring service groups using the Application Configuration Wizard”](#) on page 232.

Deleting resources from a service group

This topic describes how to delete a resource within a service group using the Application Configuration Wizard.

To delete a resource

- 1 Start the Application Configuration Wizard.
 On Windows 2012 operating systems, use the **Apps** menu.
- 2 Read the text on the Welcome panel and click **Next**.
- 3 In the Wizard Options panel, click **Modify Service Group**. From the Service Groups list, select the service group containing the resource that you want to delete and click **Next**.
- 4 In the Service Group Configuration panel, click **Next**.
- 5 In the Application Options panel, click **Delete**, select the resource you want to delete, and click **Next**.
- 6 In the Warning dialog box, click **No** to retain the selected resource. Otherwise, click **Yes**.

The specified resource will be deleted when you exit the wizard after selecting the **Configure application dependency and create service group** option in the Application Options panel.

- 7 After marking the resource for deletion, you can:
 - Add additional resources to the service group.
 See [“Adding resources to a service group”](#) on page 223.
 - Modify resources in the service group.
 See [“Modifying an application service group”](#) on page 234.
 - Add VCS components to the service group.
 See [“Configuring VCS components”](#) on page 229.
 - Create the service group.
 See [“Configuring service groups using the Application Configuration Wizard”](#) on page 232.

Deleting an application service group

This topic describes steps to delete an application service group using the Application Configuration Wizard.

To delete a service group

- 1 Start the Application Configuration Wizard on a system configured to host the application service group.
On Windows 2012 operating systems, use the **Apps** menu.
- 2 Review the information in the Welcome panel and click **Next**.
- 3 In the Wizard Options panel, click **Delete service group**, select the service group to be deleted, and click **Next**.
- 4 In the Service Group Summary panel, click **Next**.
- 5 When a message appears that informs you that the wizard will run commands to delete the service group, click **Yes**.
- 6 Click **Finish**.

Configuring the service group in a non-shared storage environment

If you are using a non-shared storage configuration, you have to use the VCS MountV – VMNSDg agents to monitor your local storage. Currently, the service group configuration wizards do not support configuring these agents in the service group. You have to configure these agents manually by using the Cluster Manager (Java Console) or the VCS commands.

VCS provides templates for configuring service groups that use non-shared storage agent resources.

The Java Console templates are located in the following directory:

```
%VCS_HOME%\Templates
```

Here, `%VCS_HOME%` is the default product installation directory, typically, `C:\Program Files\Veritas\Cluster Server`.

For information about adding a service group using templates from the Java Console, refer to the *Cluster Server Administrator's Guide*.

The following steps describe how to create a service group using the Cluster Manager (Java Console).

To configure the service group in a non-shared storage environment

- 1** Open the Cluster Manager (Java Console) from **Start > All Programs > Veritas > Veritas Cluster Server** and then click **Veritas Cluster Manager - Java Console** or, on Windows Server 2012 operating systems, from the **Apps** menu.
- 2** Log on to the cluster. On the Cluster Monitor window click **File > New Cluster**, then on the New Cluster window type **localhost** in the Host name field, and then click **OK**.
- 3** Launch the service group configuration wizard. From the Cluster Explorer window menu, click **Tools > Configuration Wizard**.
- 4** On the Service Group Configuration Wizard Welcome panel, click **Next**.
- 5** Fill in the following information and then click **Next**:
 - Specify a name for the service group.
 - Select the systems for the service group. Click a system in the Available Systems box and then click the right arrow to move the systems to Systems for Service Group.
 - Leave the service group type as the default, Failover.
- 6** Click **Next** again.

- 7 In the Templates list, select the desired service group template depending on the configuration and then click **Next**.

Template name	Description
FileShareVMNSGroup	Use these templates to create a single node high availability service group that uses non-shared storage.
IIS60VMNSGroup	
MSMQVMNSGroup	
	These templates include resources for configuring MountV and VMNSDg agents.
FileShareVirtVMNSGroup	Use these templates to create a single node high availability service group in a VMware virtual environment.
IIS60VirtVMNSGroup	
MSMQVirtVMNSGroup	
	These templates includes resources for configuring MountV, VMwareDisks, and VMNSDg agents.
VvrRvgVMNSRVGGroup	Use this template to create a Volume Replicator replication service group on a single node that uses non-shared storage.

The Templates box lists 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.

- 8 Click **Next**. The wizard starts creating the service group.
- 9 After the service group is successfully created, click **Next** to edit attributes using the wizard.
- 10 The wizard lists the resources and their attributes. You must specify values for the mandatory attributes that appear in bold. The remaining resources listed in the window are preconfigured by the template and do not require editing.

To modify an attribute, do the following:

- Click the resource.
- Click the attribute to be modified.
- Click the **Edit** icon at the end of the table row.
- In the Edit Attribute dialog box, enter the attribute values.
- Click **OK**.

For details on application-specific agent attributes, refer to the application-specific agent or solutions guide.

For details on the storage and network agent attributes, refer to the *Cluster Server Bundled Agents Reference Guide*.

- 11** Click **Finish**.
- 12** Right-click the newly created service group and select **Enable Resources**.
- 13** Right-click the newly created service group, select **Online** from the context menu, and then select a system on which to bring the service group online.

If you are configuring the service group on a node at the secondary site in a DR environment, bring the service group online only after completing all the DR configuration steps.

Setting the timeout duration for which the VMNSDg agent waits for all the disks to arrive before importing the disk group

During the disk group import operation, the VMNSDg agent waits for all the disks to arrive before importing the disk group. In case there are some missing disks, the VMNSDg resource will fault after this time. The default timeout duration is 60s. This time duration can also be controlled using the following registry key:

- HKLM\Software\Veritas\VCSBundledAgents\VMNSDg\<resource_name>\DgImportWaitTimeOut
 Set the value of the DgImportWaitTimeOut key to the time duration in seconds for which the VMNSDg agent waits for all the disks to arrive before importing the disk group.

You must create this registry key manually. Perform the following steps to create and set the DgImportWaitTimeOut registry key.

Note: Incorrectly editing the registry may severely damage your system. Make a backup copy before making changes to the registry.

To configure DgImportWaitTimeOut registry parameter:

- 1** To open the Registry Editor, press **Window+R** on the desktop (opens the Run dialog box), type **regedit**, and then click **OK**.
- 2** In the registry tree (on the left), navigate to HKLM\SOFTWARE\VERITAS\VCS\BundledAgents.
- 3** Click Edit > New > Key and create a key by the name VMNSDg , if it does not exist already.

- 4 Select the VMNSDg key and click Edit > New > Key and create a key by the name <resource_name>. Here <resource_name> should be the resource name of the VMNSDg resource. The newly created registry key should look like this:
HKLM\SOFTWARE\VERITAS\VCS\BundledAgents\VMNSDg\<resource_name>
- 5 Select the key that you created in step 4 (<resource_name>) and add a DWORD type of value. The value name should be DgImportWaitTimeOut and value data should be the desired timeout duration in seconds. The value indicates the timeout interval for which the VMNSDg agent waits for all the disks to arrive.
- 6 If there are multiple service groups to be used in the non-shared storage environment, repeat steps 4 and 5 for each VMNSDg resource that is configured in the service group.
- 7 Save and exit the Registry Editor.

About the VCS Application Manager utility

VCS starts application services under the context of the respective virtual server configured in the cluster. As the Windows MMC snap-in is not aware of the virtual server configuration, it is not possible to manage the application from the MMC snap-in.

VCS provides a utility, VCS Application Manager (VAM), that allows you to manage applications in the virtual server context. You can use VAM to launch application management tools and system management tools in the virtual server context.

VAM supports the following applications:

- Microsoft Distributed Transaction Coordinator (MSDTC)
- Microsoft Message Queuing (MSMQ)

Managing applications in virtual server context

Use the following steps to start application management tools in the virtual server context using the VCS Application Manager utility.

Before you proceed, ensure that the virtual server resource (Lanman resource) configured in the application service group is online on the node where you run the VAM utility.

To manage applications in virtual server context

- 1 Start the VCS Application Manager utility.

Click **Start > All Programs > Veritas > Veritas Cluster Server > Configuration Tools > Application Manager**.

On Windows 2012 operating systems, use the **Apps** menu.

The VCS Application Manager displays a list of supported application service groups configured in the cluster. For each service group it also displays the state of the service group, the name of the virtual server resource (Lanman resource) and the corresponding management tools used for that application.

- 2 If you wish to sort applications based on their resource type, select the desired resource type from the Select the resource type drop-down list.

The following resource types are available for selection:

- MSDTC
- MSMQ

- 3 Select an application resource that is online and then click **Manage**, or double-click the resource name.

VAM launches the management tool in the virtual server context. You can then perform the desired tasks from the management tool.

For example, if you have selected an MSDTC resource, the Computer Services snap-in is launched. You can view the distributed transactions on the virtual DTC server.

To launch a different management tool than the one displayed, click the tool name in the Managed Application column and then select the available tool from the drop-down list.

[Table 7-4](#) displays the supported applications and the respective management tools that are available.

Table 7-4 VAM: applications and tools available

Application (Resource type)	Management tools available
Microsoft Distributed Transaction Coordinator (MSDTC)	Component Services You can view the distributed transactions statistics on the DTC virtual server from a node where the MSDTC resource is online.

Table 7-4 VAM: applications and tools available (*continued*)

Application (Resource type)	Management tools available
Microsoft Message Queuing (MSMQ)	Computer Management, Performance Counters You can manage MSMQ message queues on the node where the MSMQ resource is online.

About testing resource failover using virtual fire drills

Configuring high availability for a database or an application requires several infrastructure and configuration settings on multiple systems. However, cluster environments are subject to change after the initial setup. Administrators add disks, create new diskgroups and volumes, add new cluster nodes, or new NICs to upgrade and maintain the infrastructure. Keeping the cluster configuration updated with the changing infrastructure is critical.

Virtual fire drills detect discrepancies between the VCS configuration and the underlying infrastructure on a node; discrepancies that might prevent a service group from going online on a specific node.

About virtual fire drills

The virtual fire drill feature uses the Action function associated with the agent. The Action function of the supported agents are updated to support the virtual fire drill functionality—running infrastructure checks and fixing specific errors.

The infrastructure check verifies the resources defined in the VCS configuration file (main.cf) have the required infrastructure to fail over on another node. For example, an infrastructure check for the MountV resource verifies the existence of the mount point (drive letter) defined in the MountPath attribute for the resource.

You can run an infrastructure check only when the service group is online. The check verifies that the specified node is a viable failover target capable of hosting the service group.

The virtual fire drill provides an option to fix specific errors detected during the infrastructure check.

About infrastructure checks and fixes for supported agents

[Table 7-5](#) shows the infrastructure checks for different resource types.

Table 7-5 Infrastructure checks

Resource type	Infrastructure checks	Fix option
Application	Is the specified Program available? Does the specified Program have execute permissions? Does specified user exists on host? Does the same binary exist on all nodes?	
DiskGroup	Is Veritas Volume Manager licensed? Are all disks in the diskgroup visible from the host?	
IP	Does a route exist to the IP from the specified NIC?	
Mount	Does mount directory exist? Is some other filesystem mounted at the specified mount directory?	Create the mount directory.
NIC	Does the device exist on the host?	
Process	Does the specified Program exist and does it have execute permissions? Is the specified Program a binary executable? Does the same binary exist on all nodes?	

About running a virtual fire drill

You can run a virtual fire drill from the command line or from Cluster Manager (Java Console).

HA and DR configurations using InfoScale in AWS

InfoScale Enterprise lets you configure applications for high availability (HA) and disaster recovery (DR) in Amazon Web Services (AWS) cloud.

For more information on the supported use cases, read the following topics:

- See [“Failover within the same subnet of an AZ using virtual private IP”](#) on page 245.
- See [“Failover across subnets using overlay IP”](#) on page 247.
- See [“Public access to InfoScale cluster nodes in AWS using Elastic IP”](#) on page 250.
- See [“DR from on-premise cluster to AWS”](#) on page 251.

InfoScale provides the AWSIP agent and the Route53 agent to manage the network resources for cluster communication, failover, and failback. The agents monitor and manage the IP resources within cluster nodes in the AWS cloud or cluster nodes that are spread across on-premises and AWS. For more information on the agents, see the *Cluster Server Bundled Agents Reference Guide - Linux*.

Note: Veritas recommends that you configure three coordination point servers (CP servers) in three different availability zones (AZs) to manage I/O fencing.

Application data is data is stored on the Elastic Block Store (EBS) volumes that are attached to Elastic Compute Cloud (EC2) instances. In case of clusters with two or more nodes, the Flexible Storage Sharing (FSS) feature enables network sharing of the EBS volumes. Thus, application data can be shared between the cluster nodes. For more information on FSS, see the *Storage Foundation Cluster File System High Availability Administrator's Guide - Linux*.

InfoScale lets you configure replication between EC2 instances, which is further used to support various HA and DR scenarios for applications in AWS.

For more information on the supported replication configurations in AWS, see the following documents:

- *Cluster Server Administrator's Guide - Windows*
- *Volume Replicator Administrator's Guide - Windows*

Failover within the same subnet of an AZ using virtual private IP

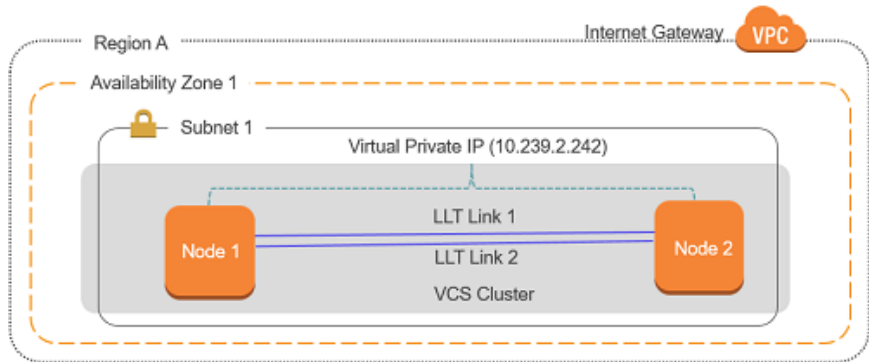
InfoScale clusters let you fail over IPs—and thereby, the application configured for HA—within the same subnet of an AZ.

The following information is required:

- The virtual private IP (secondary private IP) address to be failed over
- The device to which the IP should be plumbed
- The directory in which the AWS CLI is installed; this input is not required if it is provided in the `PATH` environment variable

Sample configuration with private IP

The following graphic depicts a sample failover configuration within the same subnet using a virtual private IP:



The sample configuration includes the following elements:

- A virtual private cloud (VPC) is configured in Region A of the AWS cloud.
- An application is configured for HA using an InfoScale cluster that comprises two nodes, Node1 and Node2, which are EC2 instances.
- Both the cluster nodes exist in the same subnet.
- A virtual private IP is configured, which is failed over from one node to the other as part of the failover or the failback operations.

Sample service group configuration with private IP

The following snippet is a service group configuration from a sample VCS configuration file (main.cf):

```
group AWSIPGrp (
    SystemList = { WIN-38PNEVJSR2K = 0 , WIN-39PNEVJSR2K = 1 }
    AutoStartList = { WIN-38PNEVJSR2K, WIN-39PNEVJSR2K }
)

AWSIP awsip (
    PrivateIP = "10.239.3.96"
    Device@WIN-38PNEVJSR2K = 12-7F-CE-5B-E2-6E
    Device@WIN-39PNEVJSR2K = 12-7F-CE-5B-E2-6F
)

IP ipres (
```

```
Address = "10.239.3.96"
SubNetMask = "255.255.254.0"
MACAddress @WIN-38PNEVJSR2K = "12:7F:CE:5B:E2:6E"
MACAddress @WIN-39PNEVJSR2K = "12:7F:CE:5B:E2:6F"
)

NIC nicres (
    MACAddress @WIN-38PNEVJSR2K = "12:7F:CE:5B:E2:6E"
    MACAddress @WIN-39PNEVJSR2K = "12:7F:CE:5B:E2:6F"
)

ipres requires nicres
awsip requires ipres
```

Failover across subnets using overlay IP

InfoScale clusters let you fail over IPs—and thereby, the application configured for HA—between different subnets in the same AZ or in different AZs.

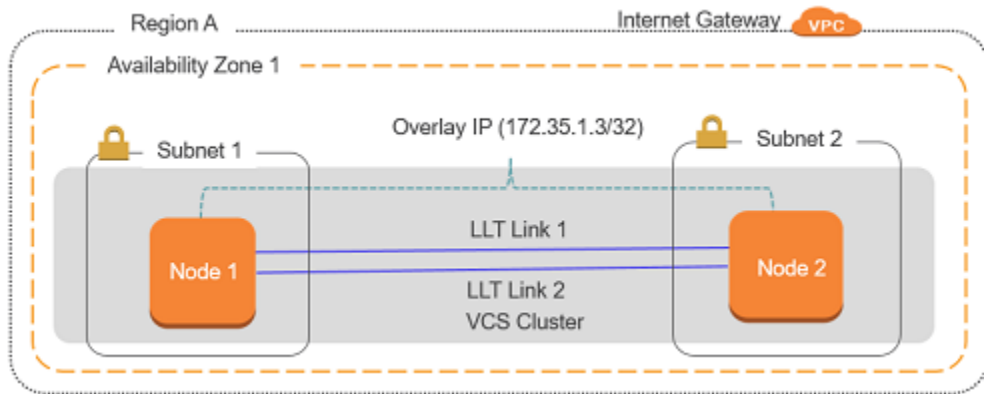
The following information is required:

- The overlay IP address to be used for failover
- The device to which the IP should be plumbed
- The directory in which the AWS CLI is installed; this input is not required if it is provided in the PATH environment variable

AWS does not allow the private IP of one subnet to be failed over to a different subnet. To overcome this limitation, InfoScale Enterprise provides an overlay IP, which is defined at the VPC level, so that it can be used across subnets.

Sample configuration with overlay IP for failover across subnets in the same AZ

The following graphic depicts a sample failover configuration across subnets within the same AZ using an overlay IP:



The sample configuration includes the following elements:

- A virtual private cloud (VPC) is configured in Region A of the AWS cloud.
- An application is configured for HA using an InfoScale cluster that comprises two nodes, Node 1 and Node 2, which are EC2 instances. Node 1 exists in Subnet 1 and Node 2 exists in Subnet 2.
- An overlay IP is configured, which allows the private IP of a node to be failed over from one subnet to another in an AZ as part of the failover or the failback operations.

Sample service group configuration with overlay IP for failover across subnets in the same AZ

The following snippet is a service group configuration from a sample VCS configuration file (main.cf):

```
group AWSIPGrp (
    SystemList = { WIN-38PNEVJSR2K = 0 , WIN-39PNEVJSR2K = 1 }
    AutoStartList = { WIN-38PNEVJSR2K, WIN-39PNEVJSR2K }
)

AWSIP overlay (
    OverlayIP = "172.16.8.55/32"
    Device @WIN-38PNEVJSR2K = 12-7F-CE-5B-E2-6E
    Device@WIN-39PNEVJSR2K = 12-7F-CE-5B-E2-6F
    RouteTableIds = { rtb-c5272ca3, rtb-fb97ac9d }
)

IP ipres (
    Address= "172.16.8.55"
    SubNetMask = "255.255.254.0"
```



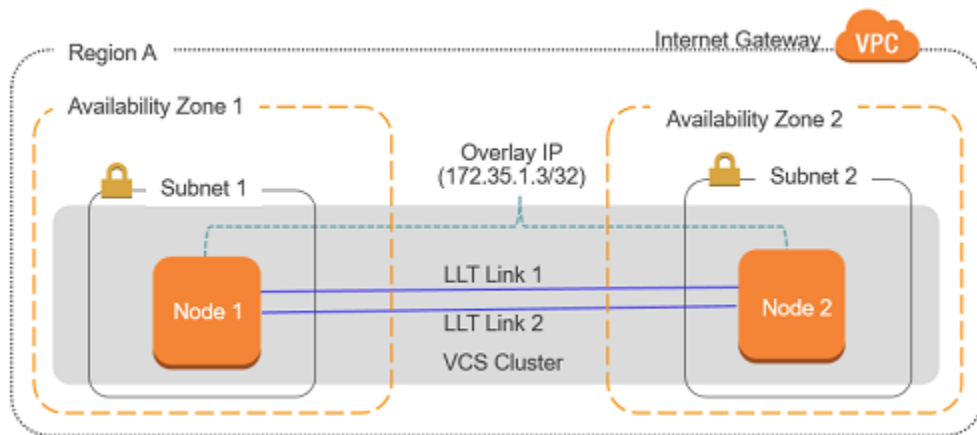
```
MACAddress @WIN-38PNEVJSR2K = "12:7F:CE:5B:E2:6E"
MACAddress @WIN-39PNEVJSR2K = "12:7F:CE:5B:E2:6F"
)
```

```
NIC nicres (
    MACAddress @WIN-38PNEVJSR2K = "12:7F:CE:5B:E2:6E"
    MACAddress @WIN-39PNEVJSR2K = "12:7F:CE:5B:E2:6F"
)
```

```
ipres requires nicres
overlayip requires ipres
```

Sample configuration with overlay IP for failover across subnets in different AZs

The following graphic depicts a sample failover configuration across subnets in different AZs using an overlay IP:



Sample service group configuration with overlay IP for failover across subnets in different AZs

The following snippet is a service group configuration from a sample VCS configuration file (main.cf):

```
group AWSIPGrp (
    SystemList = { WIN-38PNEVJSR2K = 0 , WIN-39PNEVJSR2K = 1 }
    AutoStartList = { WIN-38PNEVJSR2K, WIN-39PNEVJSR2K }
)

AWSIP overlay (
```

```

OverlayIP = "172.16.8.55/32"
Device @WIN-38PNEVJSR2K = 12-7F-CE-5B-E2-6E
Device@WIN-39PNEVJSR2K = 12-7F-CE-5B-E2-6F
RouteTableIds = { rtb-c5272ca3, rtb-fb97ac9d }
)
IP ipres (
    Address = "172.16.8.55"
    SubNetMask = "255.255.254.0"
    MACAddress @WIN-38PNEVJSR2K = "12:7F:CE:5B:E2:6E"
    MACAddress @WIN-39PNEVJSR2K = "12:7F:CE:5B:E2:6F"
)

NIC nicres (
    MACAddress @WIN-38PNEVJSR2K = "12:7F:CE:5B:E2:6E"
    MACAddress @WIN-39PNEVJSR2K = "12:7F:CE:5B:E2:6F"
)

ipres requires nicres
overlayip requires ipres

```

Public access to InfoScale cluster nodes in AWS using Elastic IP

To allow public access to an InfoScale cluster node or to an application configured for HA or DR in AWS, specify the IP to be used in the `ElasticIP` attribute for the `AWSIP` resource. For example, if you have an application that needs to be highly available and to be accessible globally, you can use the `ElasticIP` attribute of the `AWSIP` agent to ensure both.

Sample service group configuration with Elastic IP

The following snippet is a service group configuration from a sample VCS configuration file (`main.cf`):

```

group AWSIPGrp (
    SystemList = { WIN-38PNEVJSR2K = 0 , WIN-39PNEVJSR2K = 1 }
    AutoStartList = { WIN-38PNEVJSR2K, WIN-39PNEVJSR2K }
)

AWSIP elasticIP (
    PrivateIP = "10.239.3.95"
    ElasticIP = "34.193.196.156"
    Device @WIN-38PNEVJSR2K = " 12-7F-CE-5B-E2-6E"
    Device@WIN-39PNEVJSR2K = 12-7F-CE-5B-E2-6F
)

```

```
    )  
    )  
    IP ipres (  
        Address = "10.239.3.95"  
        SubNetMask = "255.255.254.0"  
        MACAddress @WIN-38PNEVJSR2K = "12:7F:CE:5B:E2:6E"  
        MACAddress @WIN-39PNEVJSR2K = "12:7F:CE:5B:E2:6F"  
    )  
  
    NIC nicres (  
        MACAddress @WIN-38PNEVJSR2K = "12:7F:CE:5B:E2:6E"  
        MACAddress @WIN-39PNEVJSR2K = "12:7F:CE:5B:E2:6F"  
    )  
  
    ipres requires nicres  
    elasticIP requires ipres
```

DR from on-premise cluster to AWS

InfoScale Enterprise lets you use the global cluster option (GCO) for DR configurations. You can use a DR configuration to fail over applications from on-premise cluster to AWS.

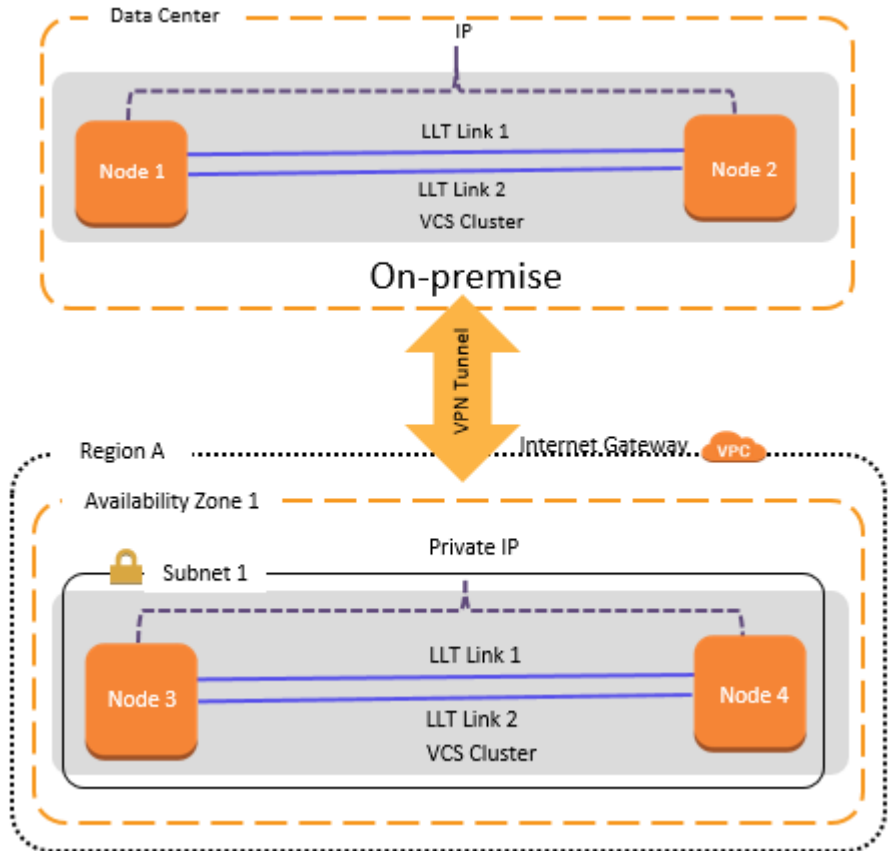
The following information is required:

- VPN tunnel information between regions or VPCs
- The Virtual private IP address to be used for cross-cluster communication.

Note: If you use an Amazon VPN tunnel in a global cluster configuration between an on-premises site and AWS, the cluster nodes in the cloud must be in the same subnet.

Sample configuration for DR across regions or VPCs

The following graphic depicts a sample DR configuration from on-premise cluster to AWS cloud:



The sample configuration includes the following elements:

- VPN tunnel between on-premise data center and Region A
- The primary site has the following elements:
 - Cluster nodes in the same subnet
 - Virtual private IP for cross-cluster communication
- The secondary site has the following elements:
 - A virtual private cloud, VPC, is configured in Region A of the AWS cloud
 - The same application is configured for HA on Node 3 and Node 4, which exist in the same subnet

Sample service group configuration for GCO from on-premise to cloud

The following snippet is a service group configuration from a sample VCS configuration file (main.cf) for a single node:

```
include "types.cf"

cluster vvr_aws (
    ClusterAddress = "10.239.3.96"
    SecureClus = 1
)

remoteclass vvr_cloud (
    ClusterAddress = "10.209.57.98"
    ConnectTimeout = 30000
    SocketTimeout = 30000
)

heartbeat Icmp (
    ClusterList = { vvr_cloud }
    AYARetryLimit = 10
    Arguments @vvr_cloud = { "10.209.57.98" }
)

system WIN-38PNEVJSR2K (
)

group ClusterService (
    SystemList = { WIN-38PNEVJSR2K = 0 }
    AutoStartList = { WIN-38PNEVJSR2K }
)

AWSIP gcoawsip (
    PrivateIP = "10.239.3.96"
    Device = 12-7F-CE-5B-E2-6E
)

IP csg_ip (
    Address = "10.239.3.96"
    SubNetMask = "255.255.254.0"
    MACAddress @WIN-38PNEVJSR2K = "12:7F:CE:5B:E2:6E"
)
```

```

NIC csg_nic (
    MACAddress @WIN-38PNEVJSR2K = "12:7F:CE:5B:E2:6E"
)

Process wac (
    StartProgram @WIN-38PNEVJSR2K = "\"C:\\Program Files\\
        Veritas\\Cluster Server\\bin\\wac.exe\""
    StopProgram @WIN-38PNEVJSR2K = "\"C:\\Program Files\\
        Veritas\\Cluster Server\\bin\\wacstop.exe\""
    MonitorProgram @WIN-38PNEVJSR2K = "\"C:\\Program Files\\
        Veritas\\Cluster Server\\bin\\wacmonitor.exe\""
)

csg_ip requires csg_nic
wac requires gcoawsip
wac requires csg_ip

group fileshare (
    SystemList = { WIN-38PNEVJSR2K = 0 }
    ClusterList = { vvr_cloud = 1, vvr_aws = 0 }
    Authority = 1
    ClusterFailOverPolicy = Auto
)

AWSIP vvrawsip (
    PrivateIP = "10.239.3.97"
    Device @WIN-38PNEVJSR2K = 12-7F-CE-5B-E2-6E
)

FileShare fileshare-FileShare (
    PathName = "\\\"
    ShareName = testshare
    LanmanResName = fileshare-Lanman
    MountResName = mounvres
    UserPermissions = { Everyone = FULL_CONTROL }
)

IP fileshare-IP (
    Address = "10.239.3.97"
    SubNetMask = "255.255.254.0"
    MACAddress @WIN-38PNEVJSR2K = 12-7F-CE-5B-E2-6E
)
    
```

```
Lanman fileshare-Lanman (
    VirtualName = FILESHARE-DHAWA
    IPResName = fileshare-IP
)

MountV mounvres (
    MountPath = "e:"
    VolumeName = datavol
    VMDGResName = vmnsdg
)

NIC fileshare_NIC (
    MACAddress @WIN-38PNEVJSR2K = "12:7F:CE:5B:E2:6E"
)

RVGPrimary rvgprimary_1 (
    RvgResourceName = vvrsg-VvrRvg
    AutoResync = 1
)

requires group vvrsg online local hard
vvrsgsip requires fileshare_NIC
fileshare-FileShare requires mounvres
fileshare-FileShare requires fileshare-Lanman
fileshare-IP requires vvrsgsip
fileshare-Lanman requires fileshare-IP
mounvres requires rvgprimary_1

group vvrsg (
    SystemList = { WIN-38PNEVJSR2K = 0 }
    AutoStartList = { WIN-38PNEVJSR2K }
)

Proxy proxy_ip (
    TargetResName = csg_ip
)

Proxy proxy_nic (
    TargetResName = csg_nic
)

VMNSDg vmnsdg (
```

```

DiskGroupName = vvrldg
DGGuid = 03d725b1-fca3-49b2-b722-38984835e6b0
)

VvrRvg vvrsg-VvrRvg (
    RVG = rvg
    VMDgResName = vmnsdg
    IPResName = proxy_ip
)

vvrsg-VvrRvg requires proxy_nic
vvrsg-VvrRvg requires proxy_ip
vvrsg-VvrRvg requires vmnsdg

```

HA and DR configurations in Azure environment

InfoScale Enterprise lets you configure applications for high availability (HA) and disaster recovery (DR) in Azure cloud environment.

For more information on the supported use cases, read the following topics:

- See [“Failover within the same subnet using a private IP”](#) on page 257.
- See [“Failover across subnets using overlay IP”](#) on page 259.
- See [“Public access to cluster nodes in Azure environment using public IP”](#) on page 261.
- See [“DR across Azure Regions or VNets and from on-premises cluster to Azure”](#) on page 262.

InfoScale Enterprise provides the AzureIP agent and the Azure DNS zone agent to manage the network resources for cluster communication, failover, and failback. These agents monitor and manage the IP resources within cluster nodes in the Azure cloud. The AzureAuth agent handles the Azure related authentication. For more information on the agents, see *Cluster Server Bundled Agents Reference Guide - Windows*.

Note: Azure recommends that you configure Azure VMs in the same Availability Set to ensure that at least one node from the cluster is always operational.

In an Azure environment, application data is stored on the Azure data disks that are attached to the Azure virtual machines. The AzureDisk agent provides high availability of the Azure disks during fail-over of an application. This agent supports high availability of managed and unmanaged disks.

VCS lets you configure replication between Azure virtual machines, which is further used to support various HA and DR scenarios for applications in Azure cloud.

For more information on the supported replication configurations in Azure, see the following documents:

- *Cluster Server Administrator's Guide - Windows*
- *Volume Replicator Administrator's Guide - Windows*

Failover within the same subnet using a private IP

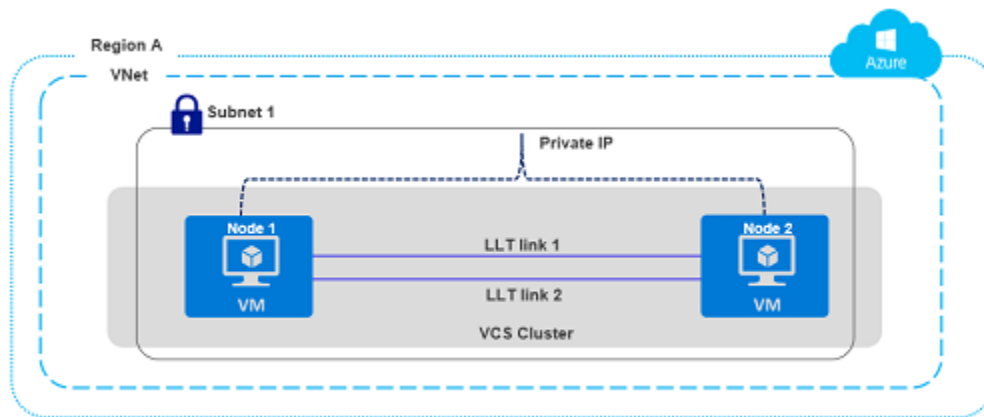
InfoScale clusters let you fail over IPs - and thereby, the application configured for HA - within the same subnet in the same VNet.

The following information is required:

- A private IP (secondary private IP) address to be failed over
- The device to which the IP should be plumbed

Sample configuration with private IP

The following graphic depicts a sample failover configuration within the same subnet using a private IP:



The sample configuration includes the following elements:

- A Azure virtual network (VNet) is configured in Region A of the Azure cloud
- An application is configured for HA using a cluster that comprises two nodes, Node1 and Node2, which are Azure virtual machines
- Both the cluster nodes exist in the same subnet
- A private IP is configured, which is failed over from one node to the other as part of the failover or the fallback operations

Sample service group configuration with private IP

The following snippet is a service group configuration from a sample configuration file (main.cf):

```
group AzureAuthGrp (
    SystemList = { AzureVM1 = 0, AzureVM2 = 1 }
    Parallel = 1
)

AzureAuth azureAuth (
    SubscriptionId = 640a326-fga6-90gh-b616-c1e9b
    ClientId = e8d899-d32a-47d04-8986-be739104d
    SecretKey = fntPgnUnhTprQrqTRonSlpRhngrrNklFn
    TenantId = 9fjkabae-2348-4308-b503-6667d61
)

Phantom phres (
)

group AzureIPGrp (
    SystemList = { AzureVM1 = 0, AzureVM2 = 1 }
)

IP IP_res (
    Address = "10.1.5.67"
    SubNetMask = "255.255.255.0"
    MACAddress @AzureVM1 = 00-0D-3A-91-73-A0
    MACAddress @AzureVM2 = 00-0D-3A-92-03-DC
)

NIC NIC_res (
    MACAddress @AzureVM1 = 00-0D-3A-91-73-A0
    MACAddress @AzureVM2 = 00-0D-3A-92-03-DC
)

AzureIP AzureIP_res (
    PrivateIP = "10.1.5.42"
    MACAddress @AzureVM1 = 00-0D-3A-91-73-A0
    MACAddress @AzureVM2 = 00-0D-3A-92-03-DC
    AzureAuthResName = azureAuth
)
```

```
IP_res requires AzureIP_res
AzureIP_res requires NIC_res
```

Failover across subnets using overlay IP

InfoScale clusters let you fail over IPs—and thereby, the application configured for HA—between different subnets in the same VNet.

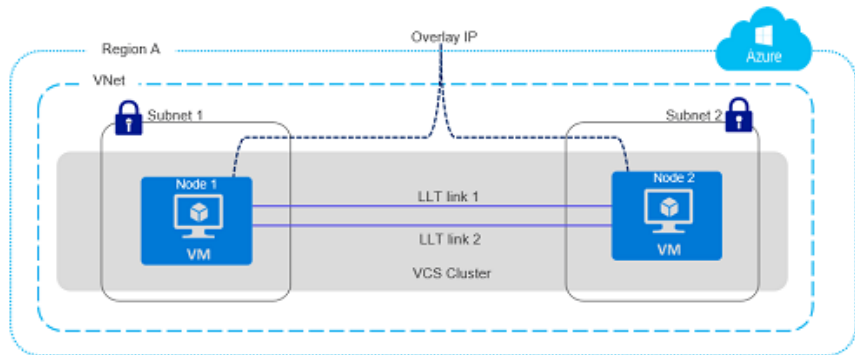
The following information is required:

- The IP address outside VNet to be used for failover
- The device to which the IP should be plumbed

Azure does not allow the private IP of one subnet to be failed over to a different subnet. To overcome this limitation, provide an overlay IP, which is outside the VNet level, so that it can be used across subnets.

Sample configuration with Overlay IP for failover across subnets in the same VNet

The following graphic depicts a sample failover configuration across subnets within the same VNet using an overlay IP:



The sample configuration includes the following elements:

- A Azure virtual network (VNet) is configured in Region A of the Azure cloud
- An application is configured for HA using a cluster that comprises two nodes, Node1 and Node2, which are Azure virtual machines
- Node 1 exists in Subnet 1 and Node 2 exists in Subnet 2.
- An overlay IP is configured to allow redirecting IP address traffic to another cluster node belonging to different subnet within the same VNet as a part of the failover or the failback operations

Sample service group configuration with overlay IP for failover across subnets in the same VNet

The following snippet is a service group configuration from a sample VCS configuration file (main.cf):

```
group AzureAuthGrp (
    SystemList = { AzureVM1 = 0, AzureVM2 = 1 }
    Parallel = 1
)

AzureAuth azureAuth (
    SubscriptionId = 640a326-fga6-90gh-b616-c1e9bb
    ClientId = e8d899-d32a-47d04-8986-be739104d
    SecretKey = fntPgnUnhTprQrqpINtrItpRhngRrNklFngLs
    TenantId = 9fjkkabae-2348-4308-b503-6667d61
)

Phantom phres (
)

group AzureOverlayIPGrp (
    SystemList = { AzureVM1 = 0, AzureVM2 = 1 }
)

IP IP_res (
    Address = "192.168.3.88"
    SubNetMask = "255.255.255.0"
    MACAddress @AzureVM1 = 00-0D-3A-91-73-A0
    MACAddress @AzureVM2 = 00-0D-3A-92-03-DC
)

NIC NIC_res (
    MACAddress @AzureVM1 = 00-0D-3A-91-73-A0
    MACAddress @AzureVM2 = 00-0D-3A-92-03-DC
)

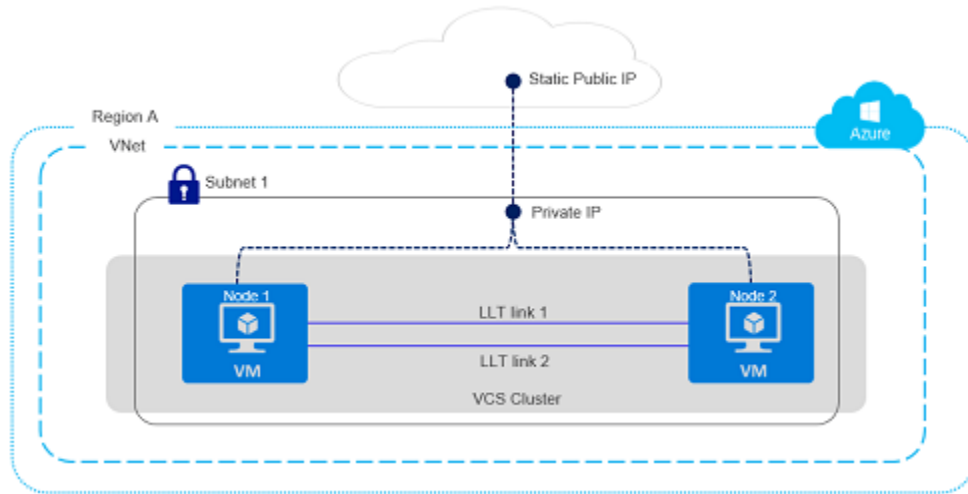
AzureIP AzureOverlayIPres (
    MACAddress @AzureVM1 = 00-0D-3A-91-73-A0
    MACAddress @AzureVM2 = 00-0D-3A-92-03-DC
    OverlayIP = "192.168.3.88"
    RouteTableResourceIds = {
        "/subscriptions/640a326-fga6-90gh-b616-c1e9bb/
        resourceGroups/TestRG/providers/Microsoft.Network/
```

```
routeTables/testRoute_eastUS2",
"/subscriptions/640a326-fga6-90gh-b616-c1e9bb/
resourceGroups/TestRG2/providers/Microsoft.Network/
routeTables/route1" }
AzureAuthResName = azureAuth
)
```

IP_res requires AzureOverlayIPres
AzureOverlayIPres requires NIC_res

Public access to cluster nodes in Azure environment using public IP

To allow public access to a cluster node or to an application configured for HA in Azure environment, specify the IP in the PublicIP attribute for the AzureIP resource. This IP is used to map the Public IP address to a secondary private IP address. For example, if you have an application that needs to be highly available and to be accessible globally, you can use the PublicIP attribute of the AzureIP agent to ensure both.



Sample service group configuration with Public IP

```
group AzureAuthGrp (
    SystemList = { AzureVM1 = 0, AzureVM2 = 1 }
    Parallel = 1
)

AzureAuth azureAuth (
    SubscriptionId = 640a326-fga6-90gh-b616-c1e9bb
```

```

        ClientId = e8d899-d32a-47d04-8986-be739104d
        SecretKey = fnlhPhrQpiNtrItpRhngrrNklFngLs
        TenantId = 9fjkaae-2348-4308-b503-6667d61
    )

    Phantom phres (

group AzurePublicIPGrp (
    SystemList = { AzureVM1 = 0, AzureVM2 = 1 }
)

    IP IP_res (
        Address = "10.1.5.67"
        SubNetMask = "255.255.255.0"
        MACAddress @AzureVM1 = 00-0D-3A-91-73-A0
        MACAddress @AzureVM2 = 00-0D-3A-92-03-DC
    )

    NIC NIC_res (
        MACAddress @AzureVM1 = 00-0D-3A-91-73-A0
        MACAddress @AzureVM2 = 00-0D-3A-92-03-DC
    )

    AzureIP AzurePublicIPres (
        PrivateIP = "10.1.5.42"
        MACAddress @AzureVM1 = 00-0D-3A-91-73-A0
        MACAddress @AzureVM2 = 00-0D-3A-92-03-DC
        PublicIP = "52.165.223.2"
        AzureAuthResName = azureAuth
    )

    IP_res requires AzurePublicIPres
    AzurePublicIPres requires NIC_res

```

DR across Azure Regions or VNets and from on-premises cluster to Azure

VCS lets you use the global cluster option (GCO) for DR configurations. You can use a DR configuration to fail over applications across different regions or VNets in Azure or between an on-premises site and Azure.

The following is required for on-premise to cloud DR using VPN tunneling:

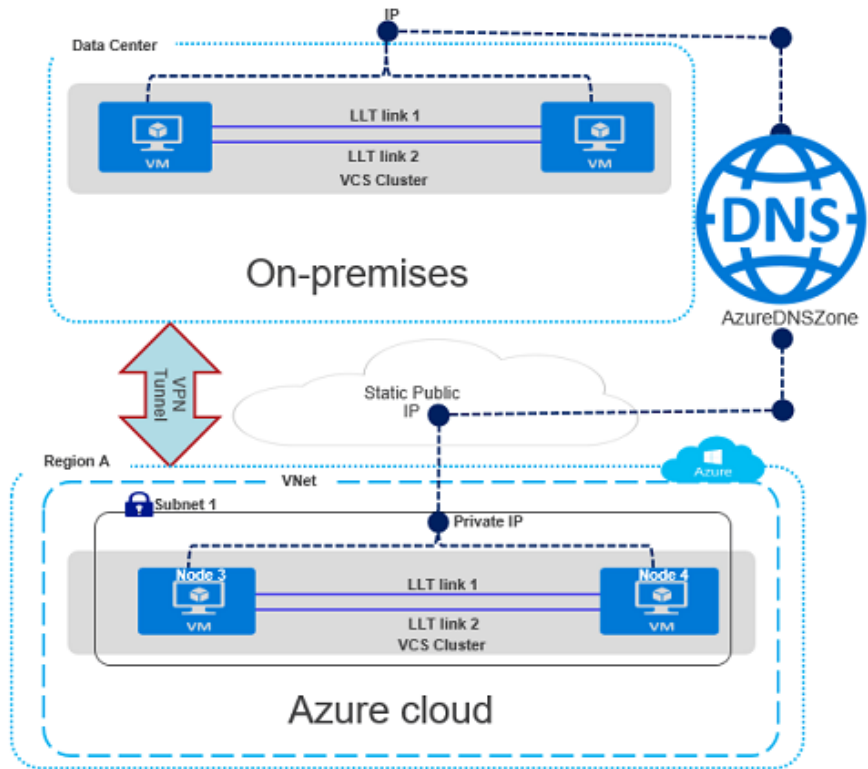
- Prepare the setup at on-premise data center
- Prepare the setup at on-cloud data center
- Establish a VPN tunnel from on-premise data center to cloud data center
- Virtual private IP for cluster nodes that exist in the same subnet. The IP address is used for cross-cluster communication

The following is required for region to region DR using VNet peering:

- Prepare the setup at both the data centers in the regions
- Establish a VNet peering from one region to the other region
- Virtual private IP for cluster nodes that exist in the same subnet. The IP address is used for cross-cluster communication

Note: If you use an VPN tunnel between an on-premises site and Azure or you use VNet peering between Azure regions, the cluster nodes in the cloud must be in the same subnet.

Sample configuration with private IP



The sample configuration includes the following elements:

- VPN tunnel between on-premise data center and Region A
- The primary site has the following elements:
 - Cluster nodes in the same subnet
 - Fencing is configured using CP servers or disk-based I/O fencing
 - Virtual private IP for cross-cluster communication
- The secondary site has the following elements:
 - A VNet is configured in Region A of the Azure cloud
 - The same application is configured for HA on Node 3 and Node 4, which exist in Subnet
 - Fencing is configured using CP servers
 - Virtual private IP for cross-cluster communication

Sample service group configuration with private IP

The following snippet is a service group configuration from a sample configuration file (main.cf):

```
cluster vcs_cluster1 (
  ClusterAddress = "10.3.3.100"
  SecureClus = 1
)

remoteclass vcs_cluster2 (
  ClusterAddress = "10.5.0.5"
)

heartbeat Icmp (
  ClusterList = { vcs_cluster2 }
  Arguments @vcs_cluster2 = { "10.5.0.5" }
)

system AzureVM1(
)
system AzureVM2(
)

group AzureAuthGrp (
  SystemList = { AzureVM1= 0, AzureVM2= 1 }
  Parallel = 1
)

AzureAuth azurauth (
  SubscriptionId = 6940a326-abc6-40dd-b717-c1e9bcd1d63
  ClientId = 8c891a8c-ghi2-473b-bbec-035ac50fb896
  SecretKey = gsiOssRooSpShuNoiQioNsjQlqHovUosQsrMt
  TenantId = 96dcased-0448-4308-b505-6789d61dd0e3
)

Phantom phres (
)

group ClusterService (
  SystemList = { AzureVM1= 0, AzureVM2= 1 }
  AutoStartList = { AzureVM1, AzureVM2}
  OnlineRetryLimit = 3
  OnlineRetryInterval = 120
)
```

)

```
Process wac (
  StartProgram @ AzureVM1 = "\"C:\\Program Files\\Veritas\\
    Cluster Server\\bin\\wac.exe\""
  StartProgram @ AzureVM2 = "\"C:\\Program Files\\Veritas\\
    Cluster Server\\bin\\wac.exe\""
  StopProgram @ AzureVM1 = "\"C:\\Program Files\\Veritas\\
    Cluster Server\\bin\\wacstop.exe\""
  StopProgram @ AzureVM2 = "\"C:\\Program Files\\Veritas\\
    Cluster Server\\bin\\wacstop.exe\""
  MonitorProgram @ AzureVM1 = "\"C:\\Program Files\\Veritas\\
    Cluster Server\\bin\\wacmonitor.exe\""
  MonitorProgram @ AzureVM2 = "\"C:\\Program Files\\Veritas\\
    Cluster Server\\bin\\wacmonitor.exe\""
```

```
AzureIP azureipres (
  PrivateIP = "10.3.3.100"
  MACAddress @AzureVM1 = 00-0D-3A-91-73-A0
  MACAddress @AzureVM2 = 00-0D-3A-92-03-DC
  AzureAuthResName = azurauth
)
```

```
IP gcoip (
  Address = "10.3.3.100"
  SubNetMask = "255.255.255.0"
  MACAddress @AzureVM1 = 00-0D-3A-91-73-A0
  MACAddress @AzureVM2 = 00-0D-3A-92-03-DC
```

)

```
NIC gconic (
  MACAddress @AzureVM1 = 00-0D-3A-91-73-A0
  MACAddress @AzureVM2 = 00-0D-3A-92-03-DC
)
```

```
gcoip requires azureipres
gcoip requires gconic
wac requires gcoip
```

```
group VVR (
  SystemList = { AzureVM1= 0, AzureVM2= 1 }
  AutoStartList = { AzureVM1, AzureVM2}
)

AzureDisk disk_res (
  DiskIds = {
    "/subscriptions/6940a326-abc6-40dd-b717-c1e9bcd6f1d63/
      resourceGroups/TestRG/providers/Microsoft.Compute/
      disks/AzureDisk1"
  }
  VMResourceGroup = TestRG
  AzureAuthResName = azurauth
)

AzureDisk disk_res1 (
  DiskIds = {
    "/subscriptions/6940a326-abc6-40dd-b717-c1e9bcd6f1d63/
      resourceGroups/TestRG/providers/Microsoft.Compute/
      disks/AzureDisk2"
  }
  VMResourceGroup = TestRG
  AzureAuthResName = azurauth
)

AzureDisk disk_res3 (
  DiskIds = {
    "/subscriptions/6940a326-abc6-40dd-b717-c1e9bcd6f1d63/
      resourceGroups/TestRG/providers/Microsoft.Compute/
      disks/AzureDisk3"
  }
  VMResourceGroup = TestRG
  AzureAuthResName = azurauth
)

AzureIP azureipres_vvr (
  PrivateIP = "10.3.3.200"
  MACAddress @AzureVM1 = 00-0D-3A-91-73-A0
  MACAddress @AzureVM2 = 00-0D-3A-92-03-DC  AzureAuthResName = azurauth
)

AzureIP azureipres_vvr1 (
  PrivateIP = "10.3.3.201"
```

```

MACAddress @AzureVM1 = 00-0D-3A-91-73-A0
MACAddress @AzureVM2 = 00-0D-3A-92-03-DC
    AzureAuthResName = azurauth
)

IP ip_vvr (
    Address = "10.3.3.200"
    SubNetMask = "255.255.255.0"
MACAddress @AzureVM1 = 00-0D-3A-91-73-A0
MACAddress @AzureVM2 = 00-0D-3A-92-03-DC
)

NIC nic_vvr (
MACAddress @AzureVM1 = 00-0D-3A-91-73-A0
MACAddress @AzureVM2 = 00-0D-3A-92-03-DC
)

VMNSDg dgres (
    DiskGroupName = vvrDg
    DGGuid = 6bee34d7-056d-4b01-ae63-397099792471
)

VvrRvg rvrgres (
    RVG = rvg
    VMDgResName = dgres
    IPResName = ip_vvr
)

ip_vvr requires azureipres_vvr
dgres requires disk_res
dgres requires disk_res1
ip_vvr requires nic_vvr
rvrgres requires ip_vvr
rvrgres requires dgres

group SG_FS (
    SystemList = { AzureVM1= 0, AzureVM2= 1 }
    ClusterList = { vcs_cluster1 = 0, vcs_cluster2 = 1 }
    Authority = 1
)

```

```
FileShare SG_FS-FileShare (
  PathName = "\\PerfLogs"
  ShareName = PerfLogs
  LanmanResName = SG_FS-Lanman
  MountResName = SG_FS-MountV
  UserPermissions = { "VCSDOMAIN\\Administrator" = FULL_CONTROL }
  ShareSubdirectories = 1
)

AzureIP SG_FS-AzureIP (
  PrivateIP = "10.3.3.300"
MACAddress @AzureVM1 = 00-0D-3A-91-73-A0
MACAddress @AzureVM2 = 00-0D-3A-92-03-DC
  AzureAuthResName = azurauth
)

IP SG_FS-IP (
  Address = "10.3.3.300"
  SubNetMask = "255.255.255.0"
MACAddress @AzureVM1 = 00-0D-3A-91-73-A0
MACAddress @AzureVM2 = 00-0D-3A-92-03-DC
)

Lanman SG_FS-Lanman (
  VirtualName = AzureVirtualFS
  IPResName = SG_FS-IP
  DNSUpdateRequired = 1
  ADUpdateRequired = 1
  DNSCriticalForOnline = 1
  ADCriticalForOnline = 1
)

MountV SG_FS-MountV (
  MountPath = "G:"
  VolumeName = volume1
  VMDGResName = dgres
)

NIC SG_FS-NIC (
MACAddress @AzureVM1 = 00-0D-3A-91-73-A0
MACAddress @AzureVM2 = 00-0D-3A-92-03-DC
)
```

```
RVGPrimary SG_FS-RVGPrimary (
  RvgResourceName = rvgres
)

requires group VVR online local hard
SG_FS-FileShare requires SG_FS-MountV
SG_FS-FileShare requires SG_FS-Lanman
SG_FS-IP requires SG_FS-NIC
SG_FS-IP requires SG_FS-AzureIP
SG_FS-Lanman requires SG_FS-IP
SG_FS-MountV requires SG_FS-RVGPrimary
```

Modifying the cluster configuration

This chapter includes the following topics:

- [About modifying the cluster configuration](#)
- [Adding nodes to a cluster](#)
- [Removing nodes from a cluster](#)
- [Reconfiguring a cluster](#)
- [Configuring single sign-on for the cluster manually](#)
- [Configuring the ClusterService group](#)
- [Deleting a cluster configuration](#)

About modifying the cluster configuration

You can modify a cluster configuration using the VCS Cluster Configuration Wizard (VCW).

When used to modify a cluster configuration, the wizard lets you perform the following tasks:

- Adds nodes to a cluster
See [“Adding nodes to a cluster”](#) on page 272.
- Remove nodes from a cluster
See [“Removing nodes from a cluster”](#) on page 276.
- Reconfigure the private network and LLT
See [“Reconfiguring a cluster”](#) on page 278.

- Configure the ClusterService service group in a cluster
See [“Configuring the ClusterService group”](#) on page 284.
- Delete a cluster

Note: When used to delete a cluster configuration, the wizard removes the cluster components from the nodes; it does not uninstall the VCS components.

See [“Deleting a cluster configuration”](#) on page 288.

You can perform the following modifications on a cluster manually:

- Configure single sign-on for the cluster
See [“Configuring single sign-on for the cluster manually”](#) on page 282.
- Convert a secure cluster to a non-secure cluster

Adding nodes to a cluster

To add a node to a VCS cluster

- 1 Start the VCS Cluster Configuration wizard.

Click **Start > All Programs > Veritas > Veritas Cluster Server > Configuration Tools > Cluster Configuration Wizard**.

Run the wizard from the node to be added or from a node in the cluster. The node that is being added should be part of the domain to which the cluster belongs.

- 2 Read the information on the Welcome panel and click **Next**.
- 3 On the Configuration Options panel, click **Cluster Operations** and click **Next**.
- 4 In the Domain Selection panel, select or type the name of the domain in which the cluster resides and select the discovery options.

To discover information about all the systems and users in the domain, do the following:

- Clear the **Specify systems and users manually** check box.
- Click **Next**.
Proceed to step [8](#).

To specify systems and user names manually (recommended for large domains), do the following:

- Check the **Specify systems and users manually** check box.

Additionally, you may instruct the wizard to retrieve a list of systems and users in the domain by selecting appropriate check boxes.

- Click **Next**.
If you chose to retrieve the list of systems, proceed to step 6. Otherwise proceed to the next step.

5 On the System Selection panel, complete the following and click **Next**:

- Type the name of an existing node in the cluster and click **Add**.
- Type the name of the system to be added to the cluster and click **Add**.

If you specify only one node of an existing cluster, the wizard discovers all nodes for that cluster. To add a node to an existing cluster, you must specify a minimum of two nodes; one that is already a part of a cluster and the other that is to be added to the cluster.

Proceed to step 8.

6 On the System Selection panel, specify the systems to be added and the nodes for the cluster to which you are adding the systems.

Enter the system name and click **Add** to add the system to the **Selected Systems** list. Alternatively, you can select the systems from the **Domain Systems** list and click the right-arrow icon.

If you specify only one node of an existing cluster, the wizard discovers all nodes for that cluster. To add a node to an existing cluster, you must specify a minimum of two nodes; one that is already a part of a cluster and the other that is to be added to the cluster.

7 The System Report panel displays the validation status, whether Accepted or Rejected, of all the systems you specified earlier.

A system can be rejected for any of the following reasons:

- The system does not respond to a ping request.
- WMI access is disabled on the system.
- The wizard is unable to retrieve information about the system's architecture or operating system.
- VCS is either not installed on the system or the version of VCS is different from what is installed on the system on which you are running the wizard.

Click on a system name to see the validation details. If you wish to include a rejected system, rectify the error based on the reason for rejection and then run the wizard again.

Click **Next** to proceed.

- 8 On the Cluster Configuration Options panel, click **Edit Existing Cluster** and click **Next**.
- 9 On the Cluster Selection panel, select the cluster to be edited and click **Next**.
If you chose to specify the systems manually in step 4, only the clusters configured with the specified systems are displayed.

- 10 On the Edit Cluster Options panel, click **Add Nodes** and click **Next**.
In the Cluster User Information dialog box, type the user name and password for a user with administrative privileges to the cluster and click **OK**.

The Cluster User Information dialog box appears only when you add a node to a cluster with VCS user privileges (a cluster that is not a secure cluster).

- 11 On the Cluster Details panel, check the check boxes next to the systems to be added to the cluster and click **Next**.

The right pane lists nodes that are part of the cluster. The left pane lists systems that can be added to the cluster.

- 12 The wizard validates the selected systems for cluster membership. After the nodes have been validated, click **Next**.

If a node does not get validated, review the message associated with the failure and restart the wizard after rectifying the problem.

- 13 On the Private Network Configuration panel, configure the VCS private network communication on each system being added and then click **Next**. How you configure the VCS private network communication depends on how it is configured in the cluster. If LLT is configured over Ethernet, you have to use the same on the nodes being added. Similarly, if LLT is configured over UDP in the cluster, you have use the same on the nodes being added.

Do one of the following:

- To configure the VCS private network over Ethernet, do the following:
 - Select the check boxes next to the two NICs to be assigned to the private network.
Veritas recommends reserving two NICs exclusively for the private network. However, you could lower the priority of one NIC and use the low-priority NIC for public and private communication.
 - If you have only two NICs on a selected system, it is recommended that you lower the priority of at least one NIC that will be used for private as well as public network communication.
To lower the priority of a NIC, right-click the NIC and select **Low Priority** from the pop-up menu.

- If your configuration contains teamed NICs, the wizard groups them as "NIC Group #N" where "N" is a number assigned to the teamed NIC. A teamed NIC is a logical NIC, formed by grouping several physical NICs together. All NICs in a team have an identical MAC address. Veritas recommends that you do not select teamed NICs for the private network. The wizard configures the LLT service (over Ethernet) on the selected network adapters.
- To configure the VCS private network over the User Datagram Protocol (UDP) layer, do the following:
 - Select the check boxes next to the two NICs to be assigned to the private network. You can assign maximum eight network links. Veritas recommends reserving at least two NICs exclusively for the VCS private network. You could lower the priority of one NIC and use the low-priority NIC for both public and private communication.
 - If you have only two NICs on a selected system, it is recommended that you lower the priority of at least one NIC that will be used for private as well as public network communication. To lower the priority of a NIC, right-click the NIC and select **Low Priority** from the pop-up menu.
 - Specify a unique UDP port for each of the link. Click **Edit Ports** if you wish to edit the UDP ports for the links. You can use ports in the range 49152 to 65535. The default ports numbers are 50000 and 50001 respectively. Click **OK**.
 - For each selected NIC, verify the displayed IP address. If a selected NIC has multiple IP addresses assigned, double-click the field and choose the desired IP address from the drop-down list. In case of IPv4, each IP address can be in a different subnet.
The IP address is used for the VCS private communication over the specified UDP port.
 - For each selected NIC, double-click the respective field in the Link column and choose a link from the drop-down list. Specify a different link (Link1 or Link2) for each NIC. Each link is associated with a UDP port that you specified earlier.

The wizard configures the LLT service (over UDP) on the selected network adapters. The specified UDP ports are used for the private network communication.

- 14 On the Public Network Communication panel, select a NIC for public network communication, for each system that is being added, and then click **Next**.

This step is applicable only if you have configured the ClusterService service group, and the system being added has multiple adapters. If the system has only one adapter for public network communication, the wizard configures that adapter automatically.

- 15 Specify the credentials for the user in whose context the VCS Helper service runs.
- 16 Review the summary information and click **Add**.
- 17 The wizard starts running commands to add the node. After all commands have been successfully run, click **Finish**.

Removing nodes from a cluster

This topic describes how to remove nodes from a multiple node VCS cluster. To remove a node from a single node cluster, you must delete the cluster.

See [“Deleting a cluster configuration”](#) on page 288.

To remove nodes from a cluster

- 1 Verify that no service groups are online on the node to be removed.
- 2 Remove the node from the SystemList of all service groups.
- 3 Start the VCS Configuration wizard.

Click **Start > All Programs > Veritas > Veritas Cluster Server > Configuration Tools > Cluster Configuration Wizard**.

On Windows 2012 operating systems, use the **Apps** menu.

- 4 Read the information on the Welcome panel and click **Next**.
- 5 In the Configuration Options panel, click **Cluster Operations** and click **Next**.
- 6 In the Domain Selection panel, select or type the name of the domain in which the cluster resides and select the domain discovery options.

To discover information about all the systems and users in the domain:

- Uncheck the **Specify systems and users manually** check box.
- Click **Next**.
- Proceed to step [10](#).

To specify systems and user names manually (recommended for large domains):

- Check the **Specify systems and users manually** check box.
Additionally, you may instruct the wizard to retrieve a list of systems and users in the domain by selecting appropriate check boxes.
 - Click **Next**.
 - If you checked **Retrieve system list from domain**, proceed to step 8.
Otherwise proceed to the next step.
- 7** In the System Selection panel, type the name of the system and click **Add**.
Proceed to step 10.
- 8** In the System Selection panel, specify the systems for the cluster from which you will be removing the nodes.

Enter the system name and click **Add** to add the system to the **Selected Systems** list. Alternatively, you can select the systems from the **Domain Systems** list and click the right-arrow icon.

If you specify only one node of an existing cluster, the wizard discovers all nodes for that cluster.
- 9** The System Report panel displays the validation status, whether Accepted or Rejected, of all the systems you specified earlier.

A system can be rejected for any of the following reasons:
- The system does not respond to a ping request.
 - WMI access is disabled on the system.
 - The wizard is unable to retrieve information about the system's architecture or operating system.
 - VCS is either not installed on the system or the version of VCS is different from what is installed on the system on which you are running the wizard.
- Click on a system name to see the validation details. If you wish to include a rejected system, rectify the error based on the reason for rejection and then run the wizard again.

Click **Next** to proceed.
- 10** In the Cluster Configuration Options panel, click **Edit Existing Cluster** and then click **Next**.
- 11** In the Cluster Selection panel, select the cluster to be edited and click **Next**.

If you chose to specify the systems manually in step 6, only the clusters configured with the specified systems are displayed.

- 12 In the Edit Cluster Options panel, click **Remove Nodes** and then click **Next**.
 In the Cluster User Information panel, enter the user name and password for a user with administrative privileges to the cluster and click **OK**.
 The Cluster User Information dialog box appears only when you remove a node from a non-secure cluster.
- 13 In the Cluster Details panel, select the check boxes next to the nodes to be removed and click **Next**.
 See [“Reconfiguring a cluster”](#) on page 278.
- 14 If you want to remove the VCS Helper service user account from the administrative group of the nodes being removed from the cluster, click **Yes** from the informational dialog box. Otherwise, click **No**.
- 15 The wizard validates the selected nodes. After the nodes have been validated, click **Next**. If a node does not get validated, review the message associated with the failure and restart the wizard after rectifying the problem.
 An informational dialog box appears if you are removing all but one nodes of a multiple node cluster. In the dialog box, specify whether you want to retain or remove the private link heartbeat.
- 16 Review the summary information and click **Remove**.
 The wizard starts running commands to remove the node from the cluster.
- 17 After the commands have been successfully run, click **Finish**.

Reconfiguring a cluster

You may need to reconfigure your cluster after changing an adapter on a cluster node, to update the LLT information, or to configure Veritas Security Services.

To reconfigure a cluster

- 1 Start the VCS Configuration wizard.
 Click **Start > All Programs > Veritas > Veritas Cluster Server > Configuration Tools > Cluster Configuration Wizard**.
 On Windows 2012 operating systems, use the **Apps** menu.
- 2 Read the information on the Welcome panel and click **Next**.
- 3 In the Configuration Options panel, click **Cluster Operations** and click **Next**.
- 4 In the Domain Selection panel, select or type the name of the domain in which the cluster resides and click **Next**.
 To discover information about all the systems and users in the domain

- Uncheck the **Specify systems and users manually** check box.
- Click **Next**.
- Proceed to step 8.

To specify systems and user names manually (recommended for large domains)

- Check the **Specify systems and users manually** check box.
Additionally, you may instruct the wizard to retrieve a list of systems and users in the domain by selecting appropriate check boxes.
- Click **Next**.
- If you checked **Retrieve system list from domain**, proceed to step 6.
Otherwise proceed to the next step.

- 5 In the System Selection panel, type the name of the system and click **Add**.
Proceed to step 8.

- 6 In the System Selection panel, specify the systems for the cluster to be reconfigured.

Enter the system name and click **Add** to add the system to the Selected Systems list. Alternatively, you can select the systems from the Domain Systems list and click the right-arrow icon.

If you specify only one node of an existing cluster, the wizard discovers all nodes for that cluster.

- 7 The System Report panel displays the validation status, whether Accepted or Rejected, of all the systems you specified earlier.

A system can be rejected for any of the following reasons:

- The system does not respond to a ping request.
- WMI access is disabled on the system.
- The wizard is unable to retrieve information about the system's architecture or operating system.
- VCS is either not installed on the system or the version of VCS is different from what is installed on the system on which you are running the wizard.

Click on a system name to see the validation details. If you wish to include a rejected system, rectify the error based on the reason for rejection and then run the wizard again.

Click **Next** to proceed.

- 8 In the Cluster Configuration Options panel, click **Edit Existing Cluster** and click **Next**.

- 9 In the Cluster Selection panel, select the cluster to be reconfigured and click **Next**. If you chose to specify the systems manually in step 4, only the clusters configured with the specified systems are displayed.
- 10 In the Edit Cluster Options panel, click **Reconfigure** and click **Next**.

In the Cluster User Information dialog box, enter the user name and password for a user with administrative privileges to the cluster and click **OK**.

The Cluster User Information dialog box appears only when you reconfigure a non-secure cluster.
- 11 In the second Edit Cluster Options dialog box, select any of the following options and click **Next**:
 - **Change private network heartbeat links**

Select this option to change the private network heartbeat links. If the selected cluster is a single node cluster, the option is to remove the private heartbeat links.

If the cluster has more than one node, the options are to add or remove private heartbeat links.

See step 12.
 - **Change HAD Helper User account**

Select this options to change the user account for the VCS Helper service.

See step 13.
 - **Configure VCS Authentication Service**

Select this option to configure the VCS authentication service for single sign-on. Single sign-on configures a secure cluster.
- 12 If the option to change the private network heartbeat links was selected, do one of the following:
 - To configure the VCS private network over Ethernet, do the following:
 - Select the check boxes next to the two NICs to be assigned to the private network.

Veritas recommends reserving two NICs exclusively for the private network. However, you could lower the priority of one NIC and use the low-priority NIC for public and private communication.
 - If you have only two NICs on a selected system, it is recommended that you lower the priority of at least one NIC that will be used for private as well as public network communication.

To lower the priority of a NIC, right-click the NIC and select **Low Priority** from the pop-up menu.

- If your configuration contains teamed NICs, the wizard groups them as "NIC Group #N" where "N" is a number assigned to the teamed NIC. A teamed NIC is a logical NIC, formed by grouping several physical NICs together. All NICs in a team have an identical MAC address. Veritas recommends that you do not select teamed NICs for the private network.

The wizard will configure the LLT service (over Ethernet) on the selected network adapters.

- To configure the VCS private network over the User Datagram Protocol (UDP) layer, do the following:
 - Select **Configure LLT over UDP on IPv4 network** or **Configure LLT over UDP on IPv6 network** depending on how LLT is configured on the existing nodes in the cluster.
 - Select the check boxes next to the NICs to be assigned to the private network. You can assign maximum eight network links. Veritas recommends reserving two NICs exclusively for the VCS private network.
 - If you have only two NICs on a selected system, it is recommended that you lower the priority of at least one NIC that will be used for private as well as public network communication.
To lower the priority of a NIC, right-click the NIC and select **Low Priority** from the pop-up menu.
 - Specify a unique UDP port for each of the link. Click **Edit Ports** if you wish to edit the UDP ports for the links. The default ports numbers are 50000 to 50007. You can use ports in the range 49152 to 65535. Click **OK**.
 - For each selected NIC, verify the displayed IP address. If a selected NIC has multiple IP addresses assigned, double-click the field and choose the desired IP address from the drop-down list. In case of IPv4, each IP address can be in a different subnet.
The IP address is used for the VCS private communication over the specified UDP port.
 - For each selected NIC, double-click the respective field in the Link column and choose a link from the drop-down list. Specify a different link (Link1 or Link2) for each NIC. Each link is associated with a UDP port that you specified earlier.

The wizard configures the LLT service (over UDP) on the selected network adapters. The specified UDP ports are used for the private network communication.

- 13** If the option to change the VCS HAD Helper User account was selected, in the VCS Helper Service User Account dialog box, specify the name of a domain user in whose context the VCS Helper service will run.

The VCS High Availability Daemon, which runs in the context of the local system built-in account, uses the VCS Helper service user context to access the network.

- Select one of the following:
 - **Existing user**
Choose an existing user account context for the VCS Helper service.
 - **New user**
Create a new user account context for the VCS Helper service.
- Enter a valid user name for the selected account and click **Next**.
Do not append the domain name to the user name; do not enter user names as `DOMAIN\user` or `user@DOMAIN`.
- Enter a password for the selected account and click **OK**.

- 14** Review the summary information and click **Reconfigure**.

- 15** The wizard starts running commands to apply the changes. After all services have been successfully configured, click **Finish**.

Configuring single sign-on for the cluster manually

This topic describes how you can manually configure single sign-on for the cluster. In a secure cluster, the VCS Authentication Service is used to secure communication between cluster nodes and clients, including the Cluster Manager (Java Console), by using digital certificates for authentication and SSL to encrypt communication over the public network. VCS uses SSL encryption and platform-based authentication. The VCS high availability engine (HAD) and Veritas Command Server run in secure mode.

Veritas recommends that you use the Cluster Configuration Wizard (VCW) to perform this task.

See [“Reconfiguring a cluster”](#) on page 278.

To create a secure cluster manually

1 Stop VCS on all nodes:

Type the following at the command prompt on one of the cluster nodes:

```
C:\>hastop -all
```

2 Stop the Veritas Command Server service on all nodes.

Type the following at the command prompt on all the cluster nodes:

```
C:\>net stop cmdserver
```

3 On each node in the cluster, create an empty file with the name `.secure` under `%VCS_HOME%\conf\config` directory.

Here, `%VCS_HOME%` represents the VCS installation directory, typically `C:\Program Files\Veritas\Cluster Server`.

4 Start the Veritas Command Server service on all nodes.

Type the following at the command prompt on all the cluster nodes:

```
C:\>net start cmdserver
```

5 On one of the cluster nodes, set the `SecureClus` attribute to 1 in the cluster configuration file.

Set the `SecureClus` attribute to 1 in the cluster configuration file `main.cf`.

Open the configuration file `main.cf` using Notepad, and add the following line in the cluster definition:

```
SecureClus = 1
```

For example:

```
cluster VCSCluster9495 (
    UserNames = { admin = gmnFmhMjnInnLvnHmk }
    Administrators = { admin }
    SecureClus = 1
    CredRenewFrequency = 0
    CounterInterval = 5
)
```

6 Save and close the configuration file.

- 7 Start the VCS engine on the node where you modified the cluster configuration file.

Type the following on the command prompt:

```
C:\>hastart
```

- 8 Start VCS on other nodes in the cluster.

Type the following on the command prompt on one of the cluster nodes:

```
C:\>hastart -all
```

Configuring the ClusterService group

Use the VCS Configuration wizard to configure the following ClusterService service group components, if you did not configure them during the initial cluster configuration:

- Notification
- GCO Option for inter-cluster communication for global clusters

Note that the wizard allows you to configure each component only once.

To configure the ClusterService group

- 1 Start the VCS Configuration wizard.

Click **Start > All Programs > Veritas > Veritas Cluster Server > Configuration Tools > Cluster Configuration Wizard**.

On Windows 2012 operating systems, use the **Apps** menu.

- 2 Read the information on the Welcome panel and click **Next**.
- 3 In the Configuration Options panel, click **Cluster Operations** and click **Next**.
- 4 In the Domain Selection panel, select or type the name of the domain in which the cluster resides and click **Next**.

To discover information about all the systems and users in the domain

- Clear the **Specify systems and users manually** check box.
- Click **Next**.
- Proceed to step [7](#).

To specify systems and user names manually (recommended for large domains)

- Check the **Specify systems and users manually** check box.

Additionally, you may instruct the wizard to retrieve a list of systems and users in the domain by selecting appropriate check boxes.

- Click **Next**.
 - If you checked the **Retrieve system list from domain** check box, proceed to step 6. Otherwise proceed to the next step.
- 5** In the System Selection panel, type the name of the system and click **Add**.
Proceed to step 7.
- 6** In the System Selection panel, specify the systems for the cluster where you will be configuring the ClusterService group.
- Enter the system name and click **Add** to add the system to the **Selected Systems** list. Alternatively, you can select the systems from the **Domain Systems** list and click the right-arrow icon.
- If you specify only one node of an existing cluster, the wizard will discover all the nodes for that cluster.
- 7** In the Cluster Configuration Options panel, click **Edit Existing Cluster** and then click **Next**.
- 8** In the Cluster Selection panel, select the cluster to be edited and click **Next**.
- If you chose to specify the systems manually in 4, only the clusters configured with the specified systems are displayed.
- 9** In the Edit Cluster Options panel, click **Configure ClusterService Options** and then click **Next**.
- In the Cluster User Information dialog box, enter the user name and password for a user with administrative privileges to the cluster and click **OK**.
- The Cluster User Information dialog box appears only when you configure a ClusterService group in a non-secure cluster.
- 10** In the Cluster Service Components panel, select from the following components to be configured in the ClusterService service group and then click **Next**.
- Check the **Notifier Option** check box to configure notification of important events to designated recipients.
See [“Configuring notification”](#) on page 286.
 - Check the **GCO Option** check box to configure the wide-area connector (WAC) process for global clusters. The WAC process is required for inter-cluster communication.
See [“Configuring the wide-area connector process for global clusters”](#) on page 287.

Configuring notification

This topic describes how to configure the notifier resource.

To configure notification

- 1 On the Notifier Options panel, specify the mode of notification to be configured and click **Next**.

You can configure VCS to generate SNMP (V2) traps on a designated server and send emails to designated recipients in response to certain events.

- 2 If you chose to configure SNMP, specify information about the SNMP console and click **Next**.

Configure the SNMP console as follows:

- Click a field in the **SNMP Console** column and type the name or IP address of the console. The specified SNMP console must be MIB 2.0 compliant.
- Click the corresponding field in the **Severity** column and select a severity level for the console.
- Click + to add a field; click - to remove a field.
- Enter an SNMP trap port. The default value is 162.

- 3 If you chose to configure SMTP server, specify information about SMTP recipients and click **Next**.

Configure the SMTP server as follows:

- Type the name of the SMTP server.
- Click a field in the **Recipients** column and enter a recipient for notification. Enter recipients as admin@example.com.
- Click the corresponding field in the **Severity** column and select a severity level for the recipient. VCS sends messages of an equal or higher severity to the recipient.
- Click + to add fields; click - to remove a field.

- 4 On the Notifier Network Card Selection panel, specify the network information and click **Next**.

Specify the network information on the Notifier Network Card Selection panel as follows:

- If the cluster has a ClusterService service group configured, you can use the NIC resource configured in the service group or configure a new NIC resource for notification.

- If you choose to configure a new NIC resource, select a network adapter for each node in the cluster. Note that the wizard lists the public network adapters along with the adapters that were assigned a low priority.
- 5 Review the summary information and choose whether you want to bring the notification resources online when VCS is started and click **Configure**.
 - 6 If you are done with the configuration, click **Finish** to exit the wizard.

Configuring the wide-area connector process for global clusters

This topic describes how to configure wide-area connector resource for global clusters.

To configure the wide-area connector process for global clusters

- 1 On the GCO Network Selection panel, specify the network information and click **Next**.

If the cluster has a ClusterService group configured, you can use the IP address configured in the service group or configure a new IP address.

Do the following:

- To specify an existing IP address, select **Use existing IP resource** and then select the IP address from the drop-down list.
- To use a new IP address, do the following:
 - In case of IPv4, select **IPV4** and then enter the IP address and associated subnet mask. Make sure that the specified IP address has a DNS entry.
 - In case of IPv6, select **IPV6** and select the IPv6 prefix from the drop-down list.

The wizard uses the prefix and automatically generates a unique IPv6 address that is valid on the network. The IPv6 option is disabled if the network does not support IPv6.

- Select a network adapter for each node in the cluster. The wizard lists the public network adapters along with the adapters that were assigned a low priority.
- 2 Review the summary information and choose whether you want to bring the resources online when VCS starts and click **Configure**.
 - 3 Click **Finish** to exit the wizard.

The wizard does not set up a global cluster environment; it configures a resource for the wide-area connector, which is required for inter-cluster communication.

For instructions on setting up a global cluster environment:

See [“Setting up a global cluster”](#) on page 396.

Deleting a cluster configuration

This topic describes how to delete a cluster configuration.

To delete a cluster configuration

- 1 Start the VCS Configuration wizard.
Click **Start > All Programs > Veritas > Veritas Cluster Server > Configuration Tools > Cluster Configuration Wizard**.
- On Windows 2012 operating systems, use the **Apps** menu.
- 2 Read the information on the Welcome panel and click **Next**.
- 3 In the Configuration Options panel, click **Cluster Operations** and click **Next**.
- 4 In the Domain Selection panel, select or type the name of the domain in which the cluster resides and click **Next**.

To discover information about all the systems and users in the domain

- Uncheck the **Specify systems and users manually** check box.
- Click **Next**.
Proceed to step [7](#).

To specify systems and user names manually (recommended for large domains)

- Check the **Specify systems and users manually** check box.
- Additionally, you may instruct the wizard to retrieve a list of systems and users in the domain by selecting appropriate check boxes.
- Click **Next**.

If you checked the **Retrieve system list from domain** check box, proceed to step 6. Otherwise proceed to the next step.

- 5 In the System Selection panel, type the name of the system and click **Add**.
Proceed to step 7.
- 6 In the System Selection panel, specify the nodes of the cluster to be deleted.
Enter the system name and click **Add** to add the system to the Selected Systems list. Alternatively, you can select the systems from the Domain Systems list and click the right-arrow icon.

If you specify only one node of an existing cluster, VCW discovers all nodes for that cluster.
- 7 In the Cluster Configuration Options panel, click **Delete Cluster** and then click **Next**.
- 8 In the Cluster Selection panel, select the cluster whose configuration is to be deleted and click **Next**.

If you chose to specify the systems manually in step 4, only the clusters configured with the specified systems are displayed.
- 9 If you want to remove the VCS Helper service user account from the administrative group of all the nodes in the cluster, click **Yes** from the informational dialog box. Otherwise, click **No**.
- 10 In the Cluster User Information panel, enter the user name and password for a user with administrative privileges to the cluster and click **OK**.

The Cluster User Information dialog box appears only when you delete a non-secure cluster.
- 11 Review the summary information and click **Unconfigure**.
- 12 The wizard starts running commands to remove the configuration from the cluster. After all commands have been successfully run, click **Finish**.

VCW removes the cluster configuration; VCW does not unconfigure the VCS Authentication Service or uninstall the product from the systems.

Administration - Beyond the basics

- [Chapter 9. Controlling VCS behavior](#)
- [Chapter 10. The role of service group dependencies](#)
- [Chapter 11. VCS event notification](#)
- [Chapter 12. VCS event triggers](#)

Controlling VCS behavior

This chapter includes the following topics:

- [VCS behavior on resource faults](#)
- [About controlling VCS behavior at the service group level](#)
- [About controlling VCS behavior at the resource level](#)
- [Changing agent file paths and binaries](#)
- [Service group workload management](#)
- [Sample configurations depicting workload management](#)

VCS behavior on resource faults

VCS considers a resource faulted in the following situations:

- When the resource state changes unexpectedly. For example, an online resource going offline.
- When a required state change does not occur. For example, a resource failing to go online or offline when commanded to do so.

In many situations, VCS agents take predefined actions to correct the issue before reporting resource failure to the engine. For example, the agent may try to bring a resource online several times before declaring a fault.

When a resource faults, VCS takes automated actions to clean up the faulted resource. The Clean function makes sure the resource is completely shut down before bringing it online on another node. This prevents concurrency violations.

When a resource faults, VCS takes all resources dependent on the faulted resource offline. The fault is thus propagated in the service group

Critical and non-critical resources

The Critical attribute for a resource defines whether a service group fails over when the resource faults. If a resource is configured as non-critical (by setting the Critical attribute to 0) and no resources depending on the failed resource are critical, the service group will not failover. VCS takes the failed resource offline and updates the group's status to PARTIAL. The attribute also determines whether a service group tries to come online on another node if, during the group's online process, a resource fails to come online.

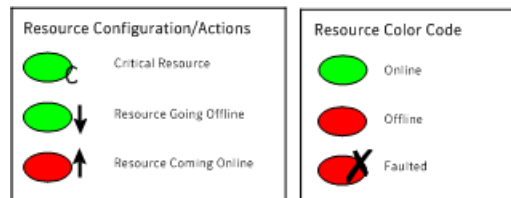
In atleast resource dependency, criticality is overridden. If the minimum criteria of atleast resource dependency is met:

- Service group does not failover even if a critical resource faults.
- Fault is not propagated up in the dependency tree even if a non-critical resource faults.
- Service group remains in ONLINE state in spite of resource faults.

VCS behavior diagrams

Figure 9-1 displays the symbols used for resource configuration and color codes.

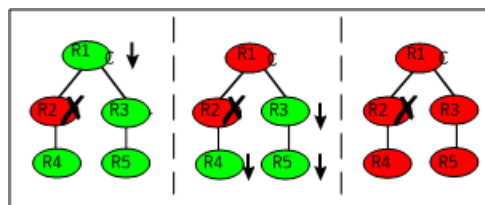
Figure 9-1 Symbols for resource configuration/actions and color codes



Example scenario 1: Resource with critical parent faults

Figure 9-2 shows an example of a service group with five resources, of which resource R1 is configured as a critical resource.

Figure 9-2 Scenario 1: Resource with critical parent faults

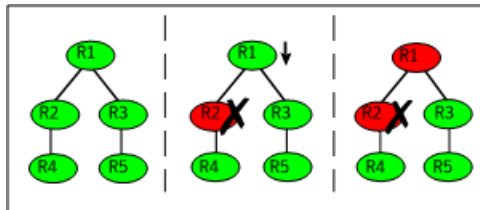


When resource R2 faults, the fault is propagated up the dependency tree to resource R1. When the critical resource R1 goes offline, VCS must fault the service group and fail it over elsewhere in the cluster. VCS takes other resources in the service group offline in the order of their dependencies. After taking resources R3, R4, and R5 offline, VCS fails over the service group to another node.

Example scenario 2: Resource with non-critical parent faults

Figure 9-3 shows an example of a service group that does not have any critical resources.

Figure 9-3 Scenario 2: Resource with non-critical parent faults

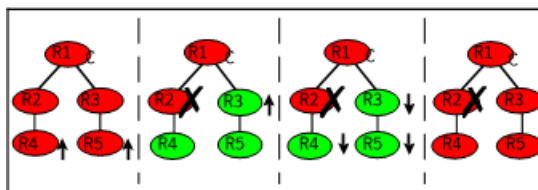


When resource R2 faults, the engine propagates the failure up the dependency tree. Neither resource R1 nor resource R2 are critical, so the fault does not result in the tree going offline or in service group failover.

Example scenario 3: Resource with critical parent fails to come online

Figure 9-4 shows an example where a command is issued to bring the service group online and resource R2 fails to come online.

Figure 9-4 Scenario 3: Resource with critical parent fails to come online



VCS calls the Clean function for resource R2 and propagates the fault up the dependency tree. Resource R1 is set to critical, so the service group is taken offline and failed over to another node in the cluster.

About controlling VCS behavior at the service group level

You can configure service group attributes to modify VCS behavior in response to resource faults.

About the AutoRestart attribute

If a persistent resource on a service group (GROUP_1) faults, VCS fails the service group over to another system if the following conditions are met:

- The AutoFailOver attribute is set.
- Another system in the cluster exists to which GROUP_1 can fail over.

If neither of these conditions is met, GROUP_1 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. If no failover targets are available, setting the AutoRestart attribute enables VCS to bring the group back online on the first available system after the group's faulted resource came online on that system.

For example, NIC is a persistent resource. In some cases, when a system boots and VCS starts, VCS probes all resources on the system. When VCS probes the NIC resource, the resource may not be online because the networking is not up and fully operational. In such situations, VCS marks the NIC resource as faulted, and does not bring the service group online. However, when the NIC resource becomes online and if AutoRestart is enabled, the service group is brought online.

About controlling failover on service group or system faults

The AutoFailOver attribute configures service group behavior in response to service group and system faults.

[Table 9-1](#) shows the possible values for the attribute AutoFailover.

Table 9-1 Possible values of the AutoFailover attribute and their description

AutoFailover attribute value	Description
0	<p>VCS does not fail over the service group when a system or service group faults.</p> <p>If a fault occurs in a service group, the group is taken offline, depending on whether any of its resources are configured as critical. If a system faults, the service group is not failed over to another system.</p>
1	<p>VCS automatically fails over the service group when a system or a service group faults, provided a suitable node exists for failover.</p> <p>The service group attributes SystemZones and FailOverPolicy impact the failover behavior of the service group. For global clusters, the failover decision is also based on the ClusterFailOverPolicy.</p> <p>See “Service group attributes” on page 546.</p>

About defining failover policies

The service group attribute FailOverPolicy governs how VCS calculates the target system for failover.

[Table 9-2](#) shows the possible values for the attribute FailoverPolicy.

Table 9-2 Possible values of the FailOverPolicy attribute and their description

FailOverPolicy attribute value	Description
Priority	<p>VCS selects the system with the lowest priority as the failover target. The Priority failover policy is ideal for simple two-node clusters or small clusters with few service groups.</p> <p>Priority is set in the SystemList attribute implicitly by ordering, such as SystemList = {SystemA, SystemB} or explicitly, such as SystemList = {SystemA=0, SystemB=1}. Priority is the default behavior.</p>
RoundRobin	<p>VCS selects the system running the fewest service groups as the failover target. This policy is ideal for large clusters running many service groups with similar server load characteristics (for example, similar databases or applications)</p>

Table 9-2 Possible values of the FailOverPolicy attribute and their description (*continued*)

FailOverPolicy attribute value	Description
Load	<p>The Load failover policy comprises the following components:</p> <p>System capacity and service group load, represented by the attributes Capacity and Load respectively.</p> <p>See System capacity and service group load on page 315.</p> <p>System limits and service group prerequisites, represented by the attributes Limits and Prerequisites, respectively.</p> <p>You cannot set the service group attribute FailOverPolicy to Load if the cluster attribute Statistics is enabled.</p> <p>See System limits and service group prerequisites on page 317.</p>
BiggestAvailable	<p>VCS selects a system based on the forecasted available capacity for all systems in the SystemList. The system with the highest forecasted available capacity is selected.</p> <p>This policy can be set only if the cluster attribute Statistics is set to Enabled. The service group attribute Load is defined in terms of CPU, Memory, and Swap in absolute units. The unit can be of the following values:</p> <ul style="list-style-type: none"> ■ CPU, MHz, or GHz for CPU ■ GB or MB for Memory ■ GB or MB for Swap <p>See “Cluster attributes” on page 583.</p>

About system zones

The SystemZones attribute enables you to create a subset of systems to use in an initial failover decision. This feature allows fine-tuning of application failover decisions, and yet retains the flexibility to fail over anywhere in the cluster.

If the attribute is configured, a service group tries to stay within its zone before choosing a host in another zone. For example, in a three-tier application infrastructure with Web, application, and database servers, you could create two system zones: one each for the application and the database. In the event of a failover, a service group in the application zone will try to fail over to another node within the zone. If no nodes are available in the application zone, the group will fail over to the database zone, based on the configured load and limits.

In this configuration, excess capacity and limits on the database backend are kept in reserve to handle the larger load of a database failover. The application servers handle the load of service groups in the application zone. During a cascading failure, the excess capacity in the cluster is available to all service groups.

Load-based autostart

VCS provides a method to determine where a service group comes online when the cluster starts. Setting the `AutoStartPolicy` to `Load` instructs the VCS engine, HAD, to determine the best system on which to start the groups. VCS places service groups in an `AutoStart` queue for load-based startup as soon as the groups probe all running systems. VCS creates a subset of systems that meet all prerequisites and then chooses the system with the highest `AvailableCapacity`.

Set `AutoStartPolicy = Load` and configure the `SystemZones` attribute to establish a list of preferred systems on which to initially run a group.

About freezing service groups

Freezing a service group prevents VCS from taking any action when the service group or a system faults. Freezing a service group prevents dependent resources from going offline when a resource faults. It also prevents the `Clean` function from being called on a resource fault.

You can freeze a service group when performing operations on its resources from outside VCS control. This prevents VCS from taking actions on resources while your operations are on. For example, freeze a database group when using database controls to stop and start a database.

About controlling Clean behavior on resource faults

The `ManageFaults` attribute specifies whether VCS calls the `Clean` function when a resource faults. `ManageFaults` can be configured at both resource level and service group level. The values of this attribute at the resource level supersede those at the service group level.

You can configure the `ManageFaults` attribute with the following possible values:

- If the `ManageFaults` attribute is set to `ALL`, VCS calls the `Clean` function when a resource faults.
- If the `ManageFaults` attribute is set to `NONE`, VCS takes no action on a resource fault; it "hangs the service group until administrative action can be taken. VCS marks the resource state as `ADMIN_WAIT` and does not fail over the service group until the resource fault is removed and the `ADMIN_WAIT` state is cleared.

VCS calls the `resadminwait` trigger when a resource enters the `ADMIN_WAIT` state due to a resource fault if the `ManageFaults` attribute is set to `NONE`. You can customize this trigger to provide notification about the fault.

When `ManageFaults` is set to `NONE` and one of the following events occur, the resource enters the `ADMIN_WAIT` state:

[Table 9-3](#) lists the possible events and the subsequent state of the resource when the `ManageFaults` attribute is set to `NONE`.

Table 9-3 Possible events when the `ManageFaults` attribute is set to `NONE`

Event	Resource state
The offline function did not complete within the expected time.	ONLINE ADMIN_WAIT
The offline function was ineffective.	ONLINE ADMIN_WAIT
The online function did not complete within the expected time.	OFFLINE ADMIN_WAIT
The online function was ineffective.	OFFLINE ADMIN_WAIT
The resource was taken offline unexpectedly.	ONLINE ADMIN_WAIT
For the online resource the monitor function consistently failed to complete within the expected time.	ONLINE MONITOR_TIMEDOUT ADMIN_WAIT

Clearing resources in the ADMIN_WAIT state

When VCS sets a resource in the `ADMIN_WAIT` state, it invokes the `resadminwait` trigger according to the reason the resource entered the state.

See [“About the `resadminwait` event trigger”](#) on page 377.

To clear a resource

- 1 Take the necessary actions outside VCS to bring all resources into the required state.
- 2 Verify that resources are in the required state by issuing the command:

```
hagrp -clearadminwait group -sys system
```

This command clears the ADMIN_WAIT state for all resources. If VCS continues to detect resources that are not in the required state, it resets the resources to the ADMIN_WAIT state.

- 3 If resources continue in the ADMIN_WAIT state, repeat step 1 and step 2, or issue the following command to stop VCS from setting the resource to the ADMIN_WAIT state:

```
hagrp -clearadminwait -fault group -sys system
```

This command has the following results:

- If the resadminwait trigger was called for the reasons 0 or 1, the resource state is set as ONLINE|UNABLE_TO_OFFLINE.
 - 0 = The offline function did not complete within the expected time.
 - 1 = The offline function was ineffective.
- If the resadminwait trigger was called for reasons 2, 3, or 4, the resource state is set as FAULTED. Note that when resources are set as FAULTED for these reasons, the clean function is not called. Verify that resources in ADMIN-WAIT are in clean, OFFLINE state prior to invoking this command.
 - 2 = The online function did not complete within the expected time.
 - 3 = The online function was ineffective.
 - 4 = The resource was taken offline unexpectedly.

When a service group has a resource in the ADMIN_WAIT state, the following service group operations cannot be performed on the resource: online, offline, switch, and flush. Also, you cannot use the hastop command when resources are in the ADMIN_WAIT state. When this occurs, you must issue the hastop command with -force option only.

About controlling fault propagation

The FaultPropagation attribute defines whether a resource fault is propagated up the resource dependency tree. It also defines whether a resource fault causes a service group failover.

You can configure the FaultPropagation attribute with the following possible values:

- If the FaultPropagation attribute is set to 1 (default), a resource fault is propagated up the dependency tree. If a resource in the path is critical, the service group is taken offline and failed over, provided the AutoFailOver attribute is set to 1.
- If the FaultPropagation is set to 0, resource faults are contained at the resource level. VCS does not take the dependency tree offline, thus preventing failover. If the resources in the service group remain online, the service group remains in the PARTIAL|FAULTED state. If all resources are offline or faulted, the service group remains in the OFFLINE| FAULTED state.

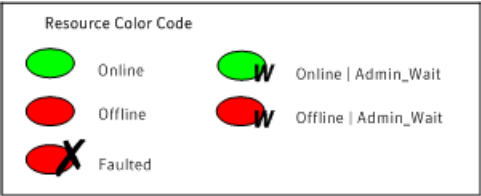
When a resource faults, VCS fires the resfault trigger and sends an SNMP trap. The trigger is called on the system where the resource faulted and includes the name of the faulted resource.

Customized behavior diagrams

This topic depicts how the ManageFaults and FaultPropagation attributes change VCS behavior when handling resource faults.

Figure 9-5 depicts the legends or resource color code.

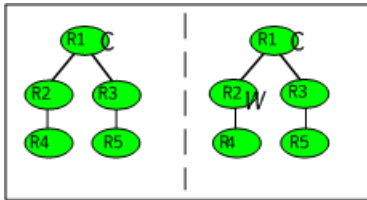
Figure 9-5 Legends and resource color code



Example scenario: Resource with a critical parent and ManageFaults=NONE

Figure 9-6 shows an example of a service group that has five resources. The ManageFaults attribute for the group of resource R2 is set to NONE.

Figure 9-6 Scenario: Resource with a critical parent and ManageFaults=NONE

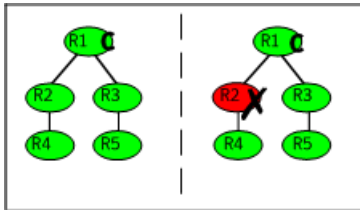


If resource R2 fails, the resource is marked as ONLINE|ADMIN_WAIT. The Clean function is not called for the resource. VCS does not take any other resource offline.

Example scenario: Resource with a critical parent and FaultPropagation=0

Figure 9-7 shows an example where the FaultPropagation attribute is set to 0.

Figure 9-7 Scenario: Resource with a critical parent and FaultPropagation=0



When resource R2 faults, the Clean function is called and the resource is marked as faulted. The fault is not propagated up the tree, and the group is not taken offline.

VCS behavior for resources that support the intentional offline functionality

Certain agents can identify when an application has been intentionally shut down outside of VCS control.

For agents that support this functionality, if an administrator intentionally shuts down an application outside of VCS control, VCS does not treat it as a fault. VCS sets the service group state as offline or partial, depending on the state of other resources in the service group.

This feature allows administrators to stop applications without causing a failover. The feature is available for V51 agents.

About the IntentionalOffline attribute

To configure a resource to recognize an intentional offline of configured application, set the IntentionalOffline attribute to 1. Set the attribute to its default value of 0 to disable this functionality. IntentionalOffline is Type level attribute and not a resource level attribute.

You can configure the IntentionalOffline attribute with the following possible values:

- If you set the attribute to 1: When the application is intentionally stopped outside of VCS control, the resource enters an OFFLINE state. This attribute does not affect VCS behavior on application failure. VCS continues to fault resources if managed corresponding applications fail.
- If you set the attribute to 0: When the application is intentionally stopped outside of VCS control, the resource enters a FAULTED state.

About the ExternalStateChange attribute

Use the ExternalStateChange attribute to control service group behavior when a configured application is intentionally started or stopped outside of VCS control.

The attribute defines how VCS handles service group state when resources are intentionally brought online or taken offline outside of VCS control.

You can configure the ExternalStateChange attribute with the values listed in [Table 9-4](#).

Table 9-4 ExternalStateChange attribute values

Attribute value	Service group behavior
OnlineGroup	If the configured application is started outside of VCS control, VCS brings the corresponding service group online. If you attempt to start the application on a frozen node or service group, VCS brings the corresponding service group online once the node or the service group is unfrozen.
OfflineGroup	If the configured application is stopped outside of VCS control, VCS takes the corresponding service group offline.
OfflineHold	<p>If a configured application is stopped outside of VCS control, the agent sets the state of the corresponding VCS resource as offline. In this case, the agents does not take any parent resource or the service group offline.</p> <p>Note: The agent sets offline state only for the resources that support detection of an intentional offline, and are registered as V51 or later agents.</p>

OfflineHold and OfflineGroup are mutually exclusive.

About controlling VCS behavior at the resource level

You can control VCS behavior at the resource level. Note that a resource is not considered faulted until the agent framework declares the fault to the VCS engine.

Certain attributes affect how the VCS agent framework reacts to problems with individual resources before informing the fault to the VCS engine.

Resource type attributes that control resource behavior

The following attributes affect how the VCS agent framework reacts to problems with individual resources before informing the fault to the VCS engine.

About the RestartLimit attribute

The RestartLimit attribute defines whether VCS attempts to restart a failed resource before it informs the engine of the fault.

If the RestartLimit attribute is set to a non-zero value, the agent attempts to restart the resource before it declares the resource as Faulted. When the agent framework restarts a failed resource, it first calls the Clean function and then the Online function. However, if the ManageFaults attribute is set to NONE, the agent does not call the Clean function and does not retry the Online function. If you configure the ManageFaults attribute at the resource level, it supersedes the values that are configured at the service group level.

About the OnlineRetryLimit attribute

The OnlineRetryLimit attribute specifies the number of times the Online function is retried if the initial attempt to bring a resource online is unsuccessful.

When OnlineRetryLimit is set to a non-zero value, the agent first calls the Clean function and then reruns the Online function. However, if the ManageFaults attribute is set to NONE, the agent does not call the Clean function and does not retry the Online function. If you configure the ManageFaults attribute at the resource level, it supersedes the values that are configured at the service group level.

About the ConflInterval attribute

The ConflInterval attribute defines how long a resource must remain online without encountering problems before previous problem counters are cleared. The attribute controls when VCS clears the RestartCount, ToleranceCount and CurrentMonitorTimeoutCount values.

About the ToleranceLimit attribute

The ToleranceLimit attribute defines the number of times the monitor routine should return an offline status before declaring a resource offline. This attribute is typically used when a resource is busy and appears to be offline. Setting the attribute to a non-zero value instructs VCS to allow multiple failing monitor cycles with the expectation that the resource will eventually respond. Setting a non-zero ToleranceLimit also extends the time required to respond to an actual fault.

About the FaultOnMonitorTimeouts attribute

The FaultOnMonitorTimeouts attribute defines whether VCS interprets a Monitor function timeout as a resource fault.

If the attribute is set to 0, VCS does not treat Monitor timeouts as a resource faults. If the attribute is set to 1, VCS interprets the timeout as a resource fault and the agent calls the Clean function to shut the resource down.

By default, the FaultOnMonitorTimeouts attribute is set to 4. This means that the Monitor function must time out four times in a row before the resource is marked faulted. The first monitor time out timer and the counter of time outs are reset after one hour of the first monitor time out.

How VCS handles resource faults

This section describes the process VCS uses to determine the course of action when a resource faults.

VCS behavior when an online resource faults

In the following example, a resource in an online state is reported as being offline without being commanded by the agent to go offline.

VCS goes through the following steps when an online resource faults:

- VCS first verifies the Monitor routine completes successfully in the required time. If it does, VCS examines the exit code returned by the Monitor routine. If the Monitor routine does not complete in the required time, VCS looks at the FaultOnMonitorTimeouts (FOMT) attribute.
- If FOMT=0, the resource will not fault when the Monitor routine times out. VCS considers the resource online and monitors the resource periodically, depending on the monitor interval.
If FOMT=1 or more, VCS compares the CurrentMonitorTimeoutCount (CMTC) with the FOMT value. If the monitor timeout count is not used up, CMTC is incremented and VCS monitors the resource in the next cycle.

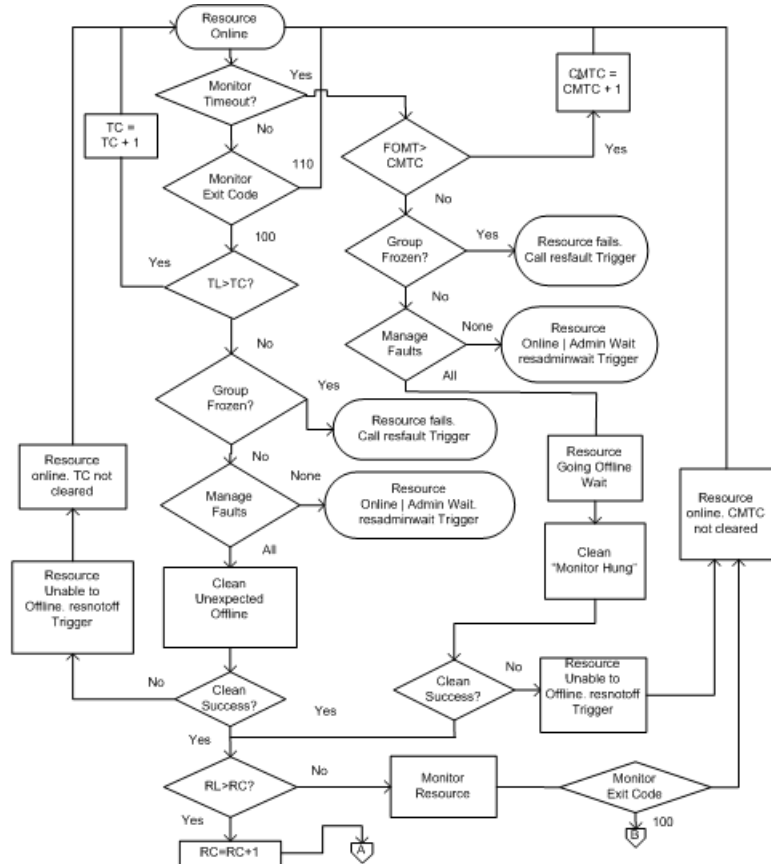
- If FOMT= CMTTC, this means that the available monitor timeout count is exhausted and VCS must now take corrective action. VCS checks the Frozen attribute for the service group. If the service group is frozen, VCS declares the resource faulted and calls the resfault trigger. No further action is taken.
- If the service group is not frozen, VCS checks the ManageFaults attribute at the resource level. VCS marks the resource as ONLINE|ADMIN_WAIT and fires the resadminwait trigger if its group-level value is NONE or if its resource-level value is IGNORE. If ManageFaults is set to ACT at the resource level or ALL at the group level, VCS invokes the Clean function with the reason Monitor Hung.

Note: The resource-level ManageFaults value supersedes the corresponding service group-level value. VCS honors the service group-level ManageFaults value only when the corresponding resource-level value is blank ("").

- If the Clean function is successful (that is, Clean exit code = 0), VCS examines the value of the RestartLimit attribute. If Clean fails (exit code = 1), the resource remains online with the state UNABLE TO OFFLINE. VCS fires the resnotoff trigger and monitors the resource again.
- If the Monitor routine does not time out, it returns the status of the resource as being online or offline.
- If the ToleranceLimit (TL) attribute is set to a non-zero value, the Monitor cycle returns offline (exit code = 100) for a number of times specified by the ToleranceLimit and increments the ToleranceCount (TC). When the ToleranceCount equals the ToleranceLimit (TC = TL), the agent declares the resource as faulted.
- If the Monitor routine returns online (exit code = 110) during a monitor cycle, the agent takes no further action. The ToleranceCount attribute is reset to 0 when the resource is online for a period of time specified by the ConflInterval attribute.
 If the resource is detected as being offline a number of times specified by the ToleranceLimit before the ToleranceCount is reset (TC = TL), the resource is considered faulted.
- After the agent determines the resource is not online, VCS checks the Frozen attribute for the service group. If the service group is frozen, VCS declares the resource faulted and calls the resfault trigger. No further action is taken.
- If the service group is not frozen, VCS checks the ManageFaults attribute. If ManageFaults=NONE, VCS marks the resource state as ONLINE|ADMIN_WAIT

and calls the `resadminwait` trigger. If `ManageFaults=ALL`, VCS calls the `Clean` function with the `CleanReason` set to `Unexpected Offline`.

- If the `Clean` function fails (exit code = 1) the resource remains online with the state `UNABLE TO OFFLINE`. VCS fires the `resnotoff` trigger and monitors the resource again. The resource enters a cycle of alternating `Monitor` and `Clean` functions until the `Clean` function succeeds or a user intervenes.
- If the `Clean` function is successful, VCS examines the value of the `RestartLimit` (RL) attribute. If the attribute is set to a non-zero value, VCS increments the `RestartCount` (RC) attribute and invokes the `Online` function. This continues till the value of the `RestartLimit` equals that of the `RestartCount`. At this point, VCS attempts to monitor the resource.
- If the `Monitor` returns an online status, VCS considers the resource online and resumes periodic monitoring. If the monitor returns an offline status, the resource is faulted and VCS takes actions based on the service group configuration.



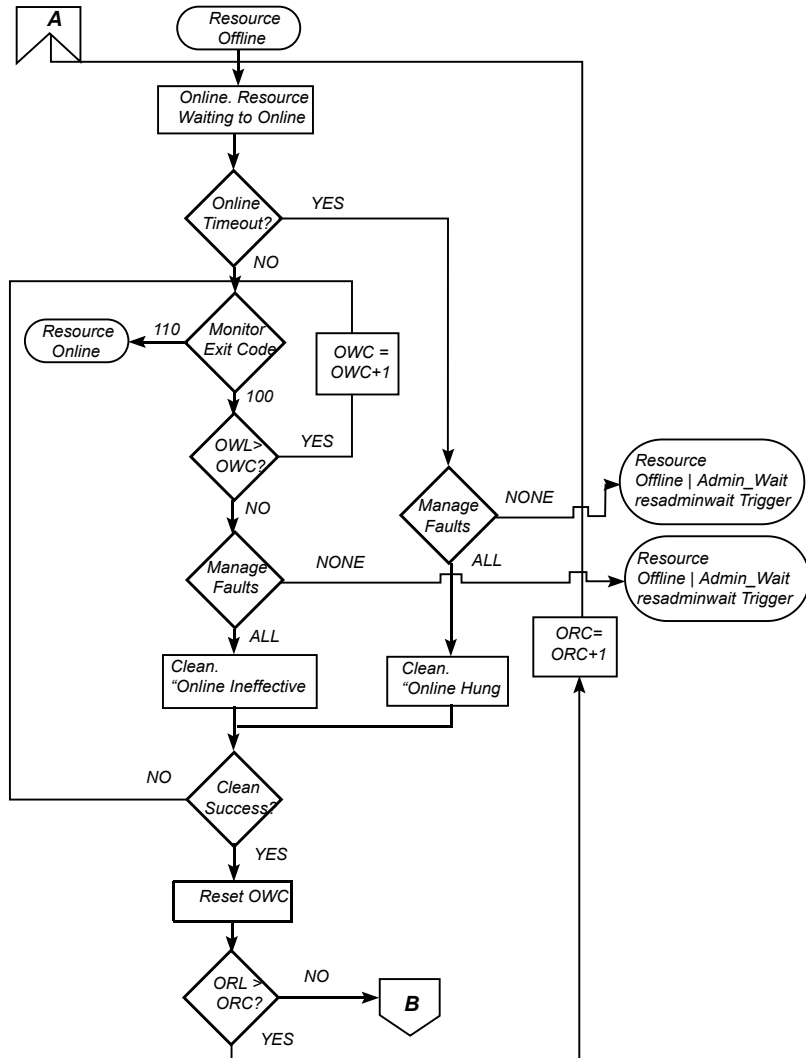
VCS behavior when a resource fails to come online

In the following example, the agent framework invokes the Online function for an offline resource. The resource state changes to WAITING TO ONLINE.

VCS goes through the following steps when a resource fails to come online:

- If the Online function times out, VCS examines the value of the ManageFaults attribute first at the resource level and then at the service group level. If ManageFaults is defined at the resource level, VCS overrides the corresponding values that are specified at the service group level. VCS takes action based on the ManageFaults values that are specified at the resource level.
- If ManageFaults is set to IGNORE at the resource level, the resource state changes to OFFLINE|ADMIN_WAIT. The resource-level value overrides the service group-level value.

- If ManageFaults is set to ACT at the resource level, VCS calls the Clean function with the CleanReason set to Online Hung.
- If resource-level ManageFaults is set to "" or blank, VCS checks the corresponding service group-level value, and proceeds as follows:
 - If ManageFaults is set to NONE, the resource state changes to OFFLINE|ADMIN_WAIT.
 - If ManageFaults is set to ALL, VCS calls the Clean function with the CleanReason set to Online Hung.
- If ManageFaults is set to NONE, the resource state changes to OFFLINE|ADMIN_WAIT.
 If ManageFaults is set to ALL, VCS calls the Clean function with the CleanReason set to Online Hung.
- If the Online function does not time out, VCS invokes the Monitor function. The Monitor routine returns an exit code of 110 if the resource is online. Otherwise, the Monitor routine returns an exit code of 100.
- VCS examines the value of the OnlineWaitLimit (OWL) attribute. This attribute defines how many monitor cycles can return an offline status before the agent framework declares the resource faulted. Each successive Monitor cycle increments the OnlineWaitCount (OWC) attribute. When OWL= OWC (or if OWL= 0), VCS determines the resource has faulted.
- VCS then examines the value of the ManageFaults attribute. If the ManageFaults is set to NONE, the resource state changes to OFFLINE|ADMIN_WAIT.
 If the ManageFaults is set to ALL, VCS calls the Clean function with the CleanReason set to Online Ineffective.
- If the Clean function is not successful (exit code = 1), the agent monitors the resource. It determines the resource is offline, and calls the Clean function with the Clean Reason set to Online Ineffective. This cycle continues till the Clean function is successful, after which VCS resets the OnlineWaitCount value.
- If the OnlineRetryLimit (ORL) is set to a non-zero value, VCS increments the OnlineRetryCount (ORC) and invokes the Online function. This starts the cycle all over again. If ORL = ORC, or if ORL = 0, VCS assumes that the Online operation has failed and declares the resource as faulted.



VCS behavior after a resource is declared faulted

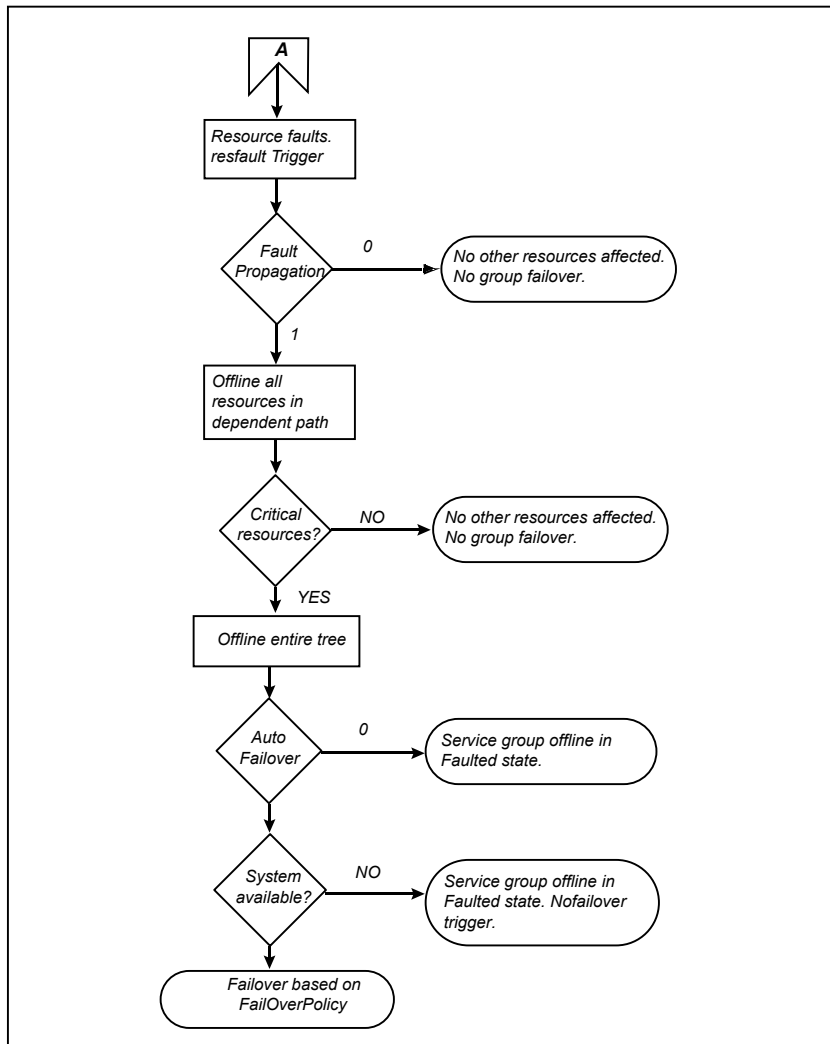
After a resource is declared faulted, VCS fires the resfault trigger and examines the value of the FaultPropagation attribute.

VCS goes through the following steps after a resource is declared faulted:

- If FaultPropagation is set to 0, VCS does not take other resources offline, and changes the group state to OFFLINE|FAULTED or PARTIAL|FAULTED. The service group does not fail over.

If FaultPropagation is set to 1, VCS takes all resources in the dependent path of the faulted resource offline, up to the top of the tree.

- VCS then examines if any resource in the dependent path is critical. If no resources are critical, the service group is left in its OFFLINE|FAULTED or PARTIAL|FAULTED state. If a resource in the path is critical, VCS takes the all resources in the service group offline in preparation of a failover.
- If the AutoFailOver attribute is set to 0, the service group is not failed over; it remains in a faulted state. If AutoFailOver is set to 1, VCS examines if any systems in the service group's SystemList are possible candidates for failover. If no suitable systems exist, the group remains faulted and VCS calls the nofailover trigger. If eligible systems are available, VCS examines the FailOverPolicy to determine the most suitable system to which to fail over the service group.
- If FailOverPolicy is set to Load, a NoFailover situation may occur because of restrictions placed on service groups and systems by Service Group Workload Management.



About 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. 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.

Use the following command to disable the resource when VCS is running:

```
# hares -modify resource_name Enabled 0
```

To have the resource disabled initially when VCS is started, set the resource's Enabled attribute to 0 in main.cf.

Limitations of disabling resources

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 for disabling resources

Following are the additional considerations for disabling resources:

- 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.
- If a service group is part of an `hagrp -online -propagate` operation or an `hagrp -offline -propagate` operation and a resource in the service group is

disabled, the resource might not complete its online operation or offline operation. In this case, PolicyIntention of the service group is set to 0.

In an `hagrp online -propagate` operation, if the child service groups cannot be brought online, the parent service groups also cannot be brought online. Therefore, when the PolicyIntention value of 1 for the child service group is cleared, the PolicyIntention value of all its parent service groups in dependency tree is also cleared. When the PolicyIntention value of 2 for the parent service group is cleared, the PolicyIntention value of all its child service groups in dependency tree is also cleared.

This section shows how a service group containing disabled resources is brought online.

Figure 9-8 shows 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.

Figure 9-8 Scenario: Transaction not propagated to the disabled resource (Resource_3)

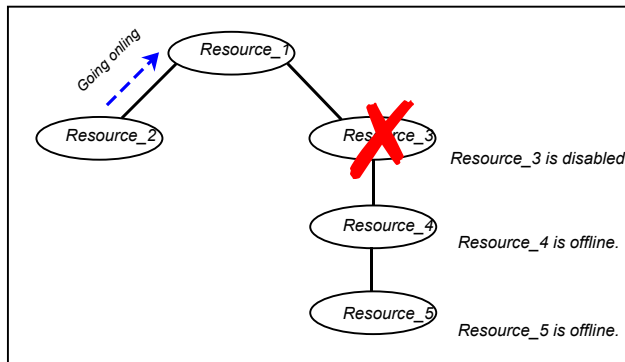
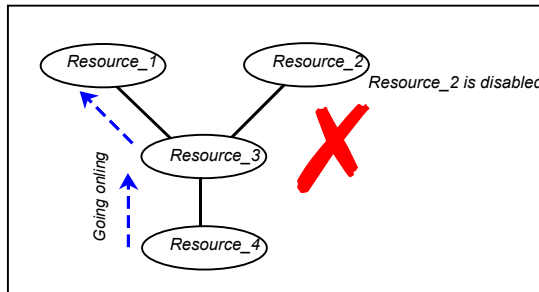


Figure 9-9, shows that Resource_2 is disabled. When the service group is brought online, resources 1, 3, 4 are also brought online (Resource_4 is brought online first). Note Resource_3, the child of the disabled resource, is brought online because Resource_1 is enabled and is dependent on it.

Figure 9-9 Scenario: Child of the disabled resource (Resource_3) is brought online



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.

Changing agent file paths and binaries

VCS runs agent binaries from the path `%VCS_HOME%\bin\agent_name\`.

You can instruct VCS to run a different set of agent binaries or scripts by specifying values for the following attributes.

- **AgentFile:**
Specify a value for this attribute if the name of the agent binary is not the same as that of the resource type.
For example, if the resource type is NetBackup and the agent binary is called NBU.dll, set the AgentFile attribute to NBU.dll.
- **AgentDirectory:**
Specify a value for this attribute if the agent is not installed at the default location.
When you specify the agent directory, VCS looks for the agent file (*AgentNameAgent*) in the agent directory. If the agent file name does not conform to the *AgentNameAgent* convention, configure the AgentFile attribute.
For example, if the NetBackup agent is installed at *C:\Program Files\VERITAS\NetBackup*, specify this path as the attribute value.

To change the path of an agent

- ◆ Before configuring a resource for the agent, set the AgentFile and AgentDirectory attributes of the agent's resource type.

```
hatype -modify resource_type AgentFile "binary_name.dll"  
hatype -modify resource_type AgentDirectory "C:\Program  
Files\agent_path"
```

Service group workload management

Workload management is a load-balancing mechanism that determines which system hosts an application during startup, or after an application or server fault.

Service Group Workload Management provides tools for making intelligent decisions about startup and failover locations, based on system capacity and resource availability.

About enabling service group workload management

The service group attribute FailOverPolicy governs how VCS calculates the target system for failover. Set FailOverPolicy to Load to enable service group workload management.

See “[About controlling VCS behavior at the resource level](#)” on page 303.

System capacity and service group load

The Load and Capacity construct allows the administrator to define a fixed amount of resources a server provides (Capacity), and a fixed amount of resources a specific service group is expected to utilize (Load).

The system attribute Capacity sets a fixed load-handling capacity for servers. Define this attribute based on system requirements.

The service group attribute Load sets a fixed demand for service groups. Define this attribute based on application requirements.

When a service group is brought online, its load is subtracted from the system's capacity to determine available capacity. VCS maintains this info in the attribute AvailableCapacity.

When a failover occurs, VCS determines which system has the highest available capacity and starts the service group on that system. During a failover involving

multiple service groups, VCS makes failover decisions serially to facilitate a proper load-based choice.

System capacity is a soft restriction; in some situations, value of the Capacity attribute could be less than zero. During some operations, including cascading failures, the value of the AvailableCapacity attribute could 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:

$\text{AvailableCapacity} = \text{Capacity} - (\text{sum of Load values of all online service groups})$

If the DynamicLoad attribute is defined, its value overrides the calculated Load values with the function:

$\text{AvailableCapacity} = \text{Capacity} - \text{DynamicLoad}$

This enables better control of system loading values than estimated service group loading (static load). However, this requires setting up and maintaining a load estimation package outside VCS. It also requires modifying the configuration file `main.cf` manually.

Note that the DynamicLoad (specified with `hasys -load`) is subtracted from the Capacity as an integer and not a percentage value. For example, if a system's capacity is 200 and the load estimation package determines the server is 80 percent loaded, it must inform VCS that the DynamicLoad value is 160 (not 80).

About 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`) for a preset time (set by the attribute `LoadTimeThreshold`), VCS invokes the loadwarning trigger.

See [“Using event triggers”](#) on page 373.

See [System attributes](#) on page 570.

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`, which determines how many seconds system load has been above `LoadWarningLevel`, is reset.

System limits and service group prerequisites

Limits is a system attribute and designates which resources are available on a system, including shared memory segments and semaphores.

Prerequisites is a service group attribute and helps manage application requirements. For example, a database may require three shared memory segments and 10 semaphores. VCS Load policy determines which systems meet the application criteria and then selects the least-loaded system.

If the prerequisites defined for a service group are not met on a system, the service group cannot be brought online on the system.

When configuring these attributes, define the service group's prerequisites first, then the corresponding system limits. Each system can have a different limit and there is no cap on the number of group prerequisites and system limits. Service group prerequisites and system limits can appear in any order.

You can also use these attributes to configure the cluster as N-to-1 or N-to-N. For example, to ensure that only one service group can be online on a system at a time, add the following entries to the definition of each group and system:

```
Prerequisites = { GroupWeight = 1 }  
Limits = { GroupWeight = 1 }
```

System limits and group prerequisites work independently of FailOverPolicy. Prerequisites determine the eligible systems on which a service group can be started. When a list of systems is created, HAD then follows the configured FailOverPolicy.

About capacity and limits

When selecting a node as a failover target, VCS selects the system that meets the service group's prerequisites and has the highest available capacity. If multiple systems meet the prerequisites and have the same available capacity, VCS selects the system appearing lexically first in the SystemList.

Systems having an available capacity of less than the percentage set by the LoadWarningLevel attribute, and those remaining at that load for longer than the time specified by the LoadTimeThreshold attribute invoke the loadwarning trigger.

Sample configurations depicting workload management

This topic lists some sample configurations that use the concepts.

System and Service group definitions

The main.cf in this example shows various Service Group Workload Management attributes in a system definition and a service group definition.

See [“About attributes and their definitions”](#) on page 524.

```
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 = 0, LargeServer2 = 1,
                  MedServer1 = 2 , MedServer2 = 3 }
    SystemZones = { LargeServer1=0, LargeServer2=0,
                   MedServer1=1, MedServer2=1 }
    AutoStartPolicy = Load
    AutoStartList = { MedServer1, MedServer2 }
    FailOverPolicy = Load
    Load = { Units = 100 }
```

```
Prerequisites = { ShrMemSeg=10, Semaphores=5, Processors=6 }  
)
```

Sample configuration: Basic four-node cluster

Following is the sample configuration for a basic four-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 = 0, Server2 = 1,  
                  Server3 = 2, Server4 = 3 }  
    AutoStartPolicy = Load  
    AutoStartList = { Server1 = 0, Server2 = 1,  
                    Server3 = 2, Server4 = 3 }  
    FailOverPolicy = Load  
    Load = { CPU = 1, Mem = 2000 }  
)  
  
group G2 (  
    SystemList = { Server1 = 0, Server2 = 1,  
                  Server3 = 2, Server4 = 3 }  
    AutoStartPolicy = Load  
    AutoStartList = { Server1, Server2, Server3, Server4 }  
    FailOverPolicy = Load  
    Load = { Units = 40 }
```

```
)

group G3 (
    SystemList = { Server1 = 0, Server2 = 1,
                   Server3 = 2, Server4 = 3 }
    AutoStartPolicy = Load
    AutoStartList = { Server1, Server2, Server3, Server4 }
    FailOverPolicy = Load
    Load = { Units = 30 }
)

group G4 (
    SystemList = { Server1 = 0, Server2 = 1,
                   Server3 = 2, Server4 = 3 }
    AutoStartPolicy = Load
    AutoStartList = { Server1, Server2, Server3, Server4 }
    FailOverPolicy = Load
    Load = { Units = 10 }
)

group G5 (
    SystemList = { Server1 = 0, Server2 = 1,
                   Server3 = 2, Server4 = 3 }
    AutoStartPolicy = Load
    AutoStartList = { Server1, Server2, Server3, Server4 }
    FailOverPolicy = Load
    Load = { Units = 50 }
)

group G6 (
    SystemList = { Server1 = 0, Server2 = 1,
                   Server3 = 2, Server4 = 3 }
    AutoStartPolicy = Load
    AutoStartList = { Server1, Server2, Server3, Server4 }
    FailOverPolicy = Load
    Load = { Units = 30 }
)

group G7 (
    SystemList = { Server1 = 0, Server2 = 1,
                   Server3 = 2, Server4 = 3 }
    AutoStartPolicy = Load
    AutoStartList = { Server1, Server2, Server3, Server4 }
```



```
FailOverPolicy = Load
Load = { Units = 20 }
)

group G8 (
    SystemList = { Server1 = 0, Server2 = 1,
                  Server3 = 2, Server4 = 3 }
    AutoStartPolicy = Load
    AutoStartList = { Server1, Server2, Server3, Server4 }
    FailOverPolicy = Load
    Load = { Units = 40 }
)
```

See [“About AutoStart operation”](#) on page 321.

About AutoStart operation

In this configuration, assume that groups probe in the same order they are described, G1 through G8. Group G1 chooses the system with the highest AvailableCapacity value. All systems have the same available capacity, so G1 starts on Server1 because this server is lexically first. Groups G2 through G4 follow on Server2 through Server4.

[Table 9-5](#) shows the Autostart cluster configuration for a basic four-node cluster with the initial four service groups online.

Table 9-5 Autostart cluster configuration for a basic four-node cluster

Server	Available capacity	Online groups
Server1	80	G1
Server2	60	G2
Server3	70	G3
Server4	90	G4

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 and G8 comes online on Server2 with AvailableCapacity of 60.

[Table 9-6](#) shows the Autostart cluster configuration for a basic four-node cluster with the other service groups online.

Table 9-6 Autostart cluster configuration for a basic four-node cluster with the other service groups online

Server	Available capacity	Online groups
Server1	50	G1 and G6
Server2	20	G2 and G8
Server3	50	G3 and G7
Server4	40	G4 and G5

In this configuration, Server2 fires the loadwarning trigger after 600 seconds because it is at the default LoadWarningLevel of 80 percent.

About the failure scenario

In the first failure scenario, Server4 fails. Group G4 chooses Server1 because Server1 and Server3 have AvailableCapacity of 50 and Server1 is lexically first. Group G5 then comes online on Server3. Serializing the failover choice allows complete load-based control and adds less than one second to the total failover time.

[Table 9-7](#) shows the cluster configuration following the first failure for a basic four-node cluster.

Table 9-7 Cluster configuration following the first failure

Server	Available capacity	Online groups
Server1	40	G1, G6, and G4
Server2	20	G2 and G8
Server3	0	G3, G7, and G5

In this configuration, Server3 fires the loadwarning trigger to notify that the server is overloaded. An administrator can then switch group G7 to Server1 to balance the load across groups G1 and G3. When Server4 is repaired, it rejoins the cluster with an AvailableCapacity value of 100, making it the most eligible target for a failover group.

About the cascading failure scenario

If Server3 fails before Server4 can be repaired, group G3 chooses Server1, group G5 chooses Server2, and group G7 chooses Server1. This results in the following configuration:

Table 9-8 shows a cascading failure scenario for a basic four node cluster.

Table 9-8 Cascading failure scenario for a basic four node cluster

Server	Available capacity	Online groups
Server1	-10	G1, G6, G4, G3, and G7
Server2	-30	G2, G8, and G5

Server1 fires the loadwarning trigger to notify that it is overloaded.

Sample configuration: Complex four-node cluster

The cluster in this example has two large enterprise servers (LargeServer1 and LargeServer2) and two medium-sized servers (MedServer1 and MedServer2). It has four service groups, G1 through 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 semaphore requirements.

```
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 = 0, LargeServer2 = 1,
               MedServer1 = 2, MedServer2 = 3 }
SystemZones = { LargeServer1=0, LargeServer2=0, MedServer1=1,
                MedServer2=1 }
AutoStartPolicy = Load
AutoStartList = { LargeServer1, LargeServer2 }
FailOverPolicy = Load
Load = { Units = 100 }
Prerequisites = { ShrMemSeg=10, Semaphores=5, Processors=6 }
)

group G2 (
SystemList = { LargeServer1 = 0, LargeServer2 = 1,
               MedServer1 = 2, MedServer2 = 3 }
SystemZones = { LargeServer1=0, LargeServer2=0, MedServer1=1,
                MedServer2=1 }
AutoStartPolicy = Load
AutoStartList = { LargeServer1, LargeServer2 }
FailOverPolicy = Load
Load = { Units = 100 }
Prerequisites = { ShrMemSeg=10, Semaphores=5, Processors=6 }
)

group G3 (
SystemList = { LargeServer1 = 0, LargeServer2 = 1,
               MedServer1 = 2, MedServer2 = 3 }
SystemZones = { LargeServer1=0, LargeServer2=0, MedServer1=1,
                MedServer2=1 }
AutoStartPolicy = Load
AutoStartList = { MedServer1, MedServer2 }
FailOverPolicy = Load
Load = { Units = 30 }
)

group G4 (
SystemList = { LargeServer1 = 0, LargeServer2 = 1,
               MedServer1 = 2, MedServer2 = 3 }
SystemZones = { LargeServer1=0, LargeServer2=0, MedServer1=1,
```

```

        MedServer2=1 }
    AutoStartPolicy = Load
    AutoStartList = { MedServer1, MedServer2 }
    FailOverPolicy = Load
    Load = { Units = 20 }
)

```

About the AutoStart operation

In this configuration, 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. If G1 probes first, it chooses LargeServer1 because LargeServer1 and LargeServer2 both have an AvailableCapacity of 200, but LargeServer1 is lexically first. Groups G3 and G4 use the same algorithm to determine their servers.

About the normal operation

[Table 9-9](#) shows the cluster configuration for a normal operation for a complex four-node cluster.

Table 9-9 Normal operation cluster configuration for a complex four-node cluster

Server	Available capacity	Current limits	Online groups
LargeServer1	100	ShrMemSeg=10 Semaphores=5 Processors=6	G1
LargeServer2	100	ShrMemSeg=10 Semaphores=5 Processors=6	G2

Table 9-9 Normal operation cluster configuration for a complex four-node cluster (*continued*)

Server	Available capacity	Current limits	Online groups
MedServer1	70	ShrMemSeg=10 Semaphores=5 Processors=6	G3
MedServer2	80	ShrMemSeg=10 Semaphores=5 Processors=6	G4

About the failure scenario

In this scenario, if LargeServer2 fails, VCS scans all available systems in group G2's SystemList that are in the same SystemZone and creates a subset of systems that meet 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:

[Table 9-10](#) shows a failure scenario cluster configuration for a complex four-node cluster.

Table 9-10 Failure scenario cluster configuration for a complex four-node cluster

Server	Available capacity	Current limits	Online groups
LargeServer1	0	ShrMemSeg=0 Semaphores=0 Processors=0	G1, G2
MedServer1	70	ShrMemSeg=10 Semaphores=5 Processors=6	G3
MedServer2	80	ShrMemSeg=10 Semaphores=5 Processors=6	G4

After 10 minutes (LoadTimeThreshold = 600) VCS fires the loadwarning trigger on LargeServer1 because the LoadWarningLevel exceeds 90 percent.

About the cascading failure scenario

In this scenario, another system failure can be tolerated because each system has sufficient Limits to accommodate the service group running on its peer. If MedServer1 fails, its groups can fail over to MedServer2.

If LargeServer1 fails, the failover of the two groups running on it is serialized. The first group lexically, G1, chooses MedServer2 because the server meets the required Limits and has AvailableCapacity value. Group G2 chooses MedServer1 because it is the only remaining system that meets the required Limits.

Sample configuration: Server consolidation

The following configuration has a complex eight-node cluster running multiple applications and 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.

In this configuration, the database zone (system zone 0) can handle a maximum of two failures. Each server has Limits to support a maximum of three database service groups. The application zone has excess capacity built into each server.

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 over across system zones and run on the least-loaded server in the application zone.

```
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 Databasel (
    SystemList = { LargeServer1 = 0, LargeServer2 = 1,
                  LargeServer3 = 2, MedServer1 = 3,
                  MedServer2 = 4, MedServer3 = 5,
                  MedServer4 = 6, MedServer5 = 6 }
    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 = { Units = 100 }
Prerequisites = { ShrMemSeg=5, Semaphores=10, Processors=6 }
)

group Database2 (
    SystemList = { LargeServer1 = 0, LargeServer2 = 1,
                  LargeServer3 = 2, MedServer1 = 3,
                  MedServer2 = 4, MedServer3 = 5,
                  MedServer4 = 6, MedServer5 = 6 }
    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 = { Units = 100 }
    Prerequisites = { ShrMemSeg=5, Semaphores=10, Processors=6 }
)

group Database3 (
    SystemList = { LargeServer1 = 0, LargeServer2 = 1,
                  LargeServer3 = 2, MedServer1 = 3,
                  MedServer2 = 4, MedServer3 = 5,
                  MedServer4 = 6, MedServer5 = 6 }
    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 = { Units = 100 }
    Prerequisites = { ShrMemSeg=5, Semaphores=10, Processors=6 }
)

group Application1 (
    SystemList = { LargeServer1 = 0, LargeServer2 = 1,
                  LargeServer3 = 2, MedServer1 = 3,
```

```
        MedServer2 = 4, MedServer3 = 5,
        MedServer4 = 6, MedServer5 = 6 }
    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 = { Units = 50 }
)

group Application2 (
    SystemList = { LargeServer1 = 0, LargeServer2 = 1,
        LargeServer3 = 2, MedServer1 = 3,
        MedServer2 = 4, MedServer3 = 5,
        MedServer4 = 6, MedServer5 = 6 }
    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 = { Units = 50 }
)

group Application3 (
    SystemList = { LargeServer1 = 0, LargeServer2 = 1,
        LargeServer3 = 2, MedServer1 = 3,
        MedServer2 = 4, MedServer3 = 5,
        MedServer4 = 6, MedServer5 = 6 }
    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 = { Units = 50 }
    )

group Application4 (
    SystemList = { LargeServer1 = 0, LargeServer2 = 1,
        LargeServer3 = 2, MedServer1 = 3,
        MedServer2 = 4, MedServer3 = 5,
        MedServer4 = 6, MedServer5 = 6 }
    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 = { Units = 50 }
)

group Application5 (
    SystemList = { LargeServer1 = 0, LargeServer2 = 1,
        LargeServer3 = 2, MedServer1 = 3,
        MedServer2 = 4, MedServer3 = 5,
        MedServer4 = 6, MedServer5 = 6 }
    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 = { Units = 50 }
)
```

About the 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

About the normal operation

[Table 9-11](#) shows the normal operation cluster configuration for a complex eight-node cluster running multiple applications and large databases.

Table 9-11 Normal operation cluster configuration for a complex eight-node cluster running multiple applications and large databases

Server	Available capacity	Current limits	Online groups
LargeServer1	100	ShrMemSeg=10 Semaphores=20 Processors=12	Database1
LargeServer2	100	ShrMemSeg=10 Semaphores=20 Processors=12	Database2
LargeServer3	100	ShrMemSeg=10 Semaphores=20 Processors=12	Database3
MedServer1	50	ShrMemSeg=5 Semaphores=10 Processors=6	Application1

Table 9-11 Normal operation cluster configuration for a complex eight-node cluster running multiple applications and large databases
(continued)

Server	Available capacity	Current limits	Online groups
MedServer2	50	ShrMemSeg=5 Semaphores=10 Processors=6	Application2
MedServer3	50	ShrMemSeg=5 Semaphores=10 Processors=6	Application3
MedServer4	50	ShrMemSeg=5 Semaphores=10 Processors=6	Application4
MedServer5	50	ShrMemSeg=5 Semaphores=10 Processors=6	Application5

About the failure scenario

In the following example, LargeServer3 fails. VCS scans all available systems in the SystemList for the Database3 group for systems in the same SystemZone and identifies systems that meet the group's prerequisites. In this case, LargeServer1 and LargeServer2 meet the required Limits. Database3 is brought online on LargeServer1. This results in the following configuration:

[Table 9-12](#) shows the failure scenario for a complex eight-node cluster running multiple applications and large databases.

Table 9-12 Failure scenario for a complex eight-node cluster running multiple applications and large databases

Server	Available capacity	Current limits	Online groups
LargeServer1	0	ShrMemSeg=5 Semaphores=10 Processors=6	Database1 Database3

Table 9-12 Failure scenario for a complex eight-node cluster running multiple applications and large databases (*continued*)

Server	Available capacity	Current limits	Online groups
LargeServer2	100	ShrMemSeg=10 Semaphores=20 Processors=12	Database2

In this scenario, further failure of either system can be tolerated because each has sufficient Limits available to accommodate the additional service group.

About the cascading failure scenario

If the performance of a database is unacceptable with two database groups running on a single server, the SystemZones policy can help expedite performance. Failing over a database group into the application zone has the effect of resetting the group's preferred zone. For example, in the above scenario Database3 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 Database3 in Zone1 (application zone). If a failure occurs in Database3, the group selects the least-loaded server in the application zone for failover.

The role of service group dependencies

This chapter includes the following topics:

- [About service group dependencies](#)
- [Service group dependency configurations](#)
- [Frequently asked questions about group dependencies](#)
- [About linking service groups](#)
- [VCS behavior with service group dependencies](#)

About service group dependencies

Service groups can be dependent on each other. The dependent group is the parent and the other group is the child. For example a finance application (parent) may require that the database application (child) is online before it comes online. While service group dependencies offer more features to manage application service groups, they create more complex failover configurations.

A service group may function both as a parent and a child. In VCS, a dependency tree may be up to five levels deep.

About dependency links

The dependency relationship between a parent and a child is called a link. The link is characterized by the dependency category, the location of the service groups, and the rigidity of dependency.

- A dependency may be online, or offline.

- A dependency may be local, global, remote, or site.
 - A dependency may be soft, firm, or hard with respect to the rigidity of the constraints between parent and child service group.
- You can customize the behavior of service groups by choosing the right combination of the dependency category, location, and rigidity

Dependency categories

Dependency categories determine the relationship of the parent group with the state of the child group.

Table 10-1 shows dependency categories and relationships between parent and child service groups.

Table 10-1 Dependency categories

Dependency	Relationship between parent and child service groups
Online group dependency	<p>The parent group must wait for the child group to be brought online before it can start.</p> <p>For example, to configure a database application and a database service as two separate groups, specify the database application as the parent, and the database service as the child.</p> <p>Online group dependency supports various location-based and rigidity-based combinations.</p>
Offline group dependency	<p>The parent group can be started only if the child group is offline and vice versa. This behavior prevents conflicting applications from running on the same system.</p> <p>For example, configure a test application as the parent and the production application as the child. The parent and child applications can be configured on the same system or on different systems.</p> <p>Offline group dependency supports only offline-local dependency.</p>

Dependency location

The relative location of the parent and child service groups determines whether the dependency between them is a local, global, remote or site.

Table 10-2 shows the dependency locations for local, global, remote and site dependencies.

Table 10-2 Dependency location

Dependency	Relative location of the parent and child service groups
Local dependency	The parent group depends on the child group being online or offline on the same system.
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 cluster.
Remote dependency	An instance of the parent group depends on one or more instances of the child group being online on any system in the cluster other than the system on which the parent is online.
Site dependency	An instance of the parent group depends on one or more instances of the child group being online on any system in the same site.

Dependency rigidity

The type of dependency defines the rigidity of the link between parent and child groups. A soft dependency means minimum constraints, whereas a hard dependency means maximum constraints.

[Table 10-3](#) shows dependency rigidity and associated constraints.

Table 10-3 Dependency rigidity

Dependency rigidity	Constraints between parent and child service groups
Soft dependency	<p>Specifies the minimum constraints while bringing parent and child groups online. The only constraint is that the child group must be online before the parent group is brought online.</p> <p>For example, in an online local soft dependency, an instance of the child group must be online on the same system before the parent group can come online.</p> <p>Soft dependency provides the following flexibility:</p> <ul style="list-style-type: none">■ If the child group faults, VCS does not immediately take the parent offline. If the child group cannot fail over, the parent remains online.■ When both groups are online, either group, child or parent, may be taken offline while the other remains online.■ If the parent group faults, the child group remains online.■ When the link is created, the child group need not be online if the parent is online. However, when both groups are online, their online state must not conflict with the type of link.

Table 10-3 Dependency rigidity (*continued*)

Dependency rigidity	Constraints between parent and child service groups
Firm dependency	<p>Imposes more constraints when VCS brings the parent or child groups online or takes them offline. In addition to the constraint that the child group must be online before the parent group is brought online, the constraints include:</p> <ul style="list-style-type: none"> ■ If the child group faults, the parent is taken offline. If the parent is frozen at the time of the fault, the parent remains in its original state. If the child cannot fail over to another system, the parent remains offline. ■ If the parent group faults, the child group may remain online. ■ The child group cannot be taken offline if the parent group is online. The parent group can be taken offline while the child is online. ■ When the link is created, the parent group must be offline. However, if both groups are online, their online state must not conflict with the type of link.
Hard dependency	<p>Imposes the maximum constraints when VCS brings the parent of child service groups online or takes them offline. For example:</p> <ul style="list-style-type: none"> ■ If a child group faults, the parent is taken offline before the child group is taken offline. If the child group fails over, the parent fails over to another system (or the same system for a local dependency). If the child group cannot fail over, the parent group remains offline. ■ If the parent faults, the child is taken offline. If the child fails over, the parent fails over. If the child group cannot fail over, the parent group remains offline. <p>Note: When the child faults, if the parent group is frozen, the parent remains online. The faulted child does not fail over.</p> <p>The following restrictions apply when configuring a hard dependency:</p> <ul style="list-style-type: none"> ■ Only online local hard dependencies are supported. ■ Only a single-level, parent-child relationship can be configured as a hard dependency. ■ A child group can have only one online hard parent group. Likewise, a parent group can have only one online hard child group. ■ Bringing the child group online does not automatically bring the parent online. ■ Taking the parent group offline does not automatically take the child offline. ■ Bringing the parent online is prohibited if the child is offline. ■ Hard dependencies are allowed only at the bottommost level of dependency tree (leaf level).

About dependency limitations

Following are some service group dependency limitations:

- A group dependency tree may be at most five levels deep.
- You cannot link two service groups whose current states violate the relationship.
 For example, all link requests are accepted if all instances of parent group are offline.
 All link requests are rejected if parent group is online and child group is offline, except in offline dependencies and in soft dependencies.
 All online global link requests, online remote link requests, and online site link requests to link two parallel groups are rejected.
 All online local link requests to link a parallel parent group to a failover child group are rejected.
- Linking service groups using site dependencies:
 - If the service groups to be linked are online on different sites, you cannot use site dependency to link them.
 - All link requests to link parallel or hybrid parent groups to a failover or hybrid child service group are rejected.
 - If two service groups are already linked using a local, site, remote, or global dependency, you must unlink the existing dependency and use site dependency. However, you can configure site dependency with other online dependencies in multiple child or multiple parent configurations.

Service group dependency configurations

In the following tables, the term instance applies to parallel groups only. If a parallel group is online on three systems, for example, an instance of the group is online on each system. For failover groups, only one instance of a group is online at any time. The default dependency type is Firm.

See [“Service group attributes”](#) on page 546.

About failover parent / failover child

[Table 10-4](#) shows service group dependencies for failover parent / failover child.

Table 10-4 Service group dependency configurations: Failover parent / Failover child

Link	Failover parent depends on ...	Failover parent is online If ...	If failover child faults, then ...	If failover parent faults, then ...
online local soft	Failover Child online on same system.	Child is online on same system.	Parent stays online. If Child fails over to another system, Parent migrates to the same system. If Child cannot fail over, Parent remains online.	Child stays online.
online local firm	Failover Child online on same system.	Child is online on same system.	Parent taken offline. If Child fails over to another system, Parent migrates to the same system. If Child cannot fail over, Parent remains offline.	Child stays online.
online local hard	Failover Child online on same system.	Child is online on same system.	Parents taken offline before Child is taken offline. If Child fails over to another system, Parent migrates to the same system. If Child cannot fail over, Parent remains offline.	Child taken offline. If Child fails over, Parent migrates to the same system. If Child cannot fail over, Parent remains offline.

Table 10-4 Service group dependency configurations: Failover parent / Failover child (*continued*)

Link	Failover parent depends on ...	Failover parent is online If ...	If failover child faults, then ...	If failover parent faults, then ...
online global soft	Failover Child online somewhere in the cluster.	Child is online somewhere in the cluster.	Parent stays online. If Child fails over to another system, Parent remains online. If Child cannot fail over, Parent remains online.	Child stays online. Parent fails over to any available system. If no failover target system is available, Parent remains offline.
online global firm	Failover Child online somewhere in the cluster.	Child is online somewhere in the cluster.	Parent taken offline after Child is taken offline. If Child fails over to another system, Parent is brought online on any system. If Child cannot fail over, Parent remains offline.	Child stays online. Parent fails over to any available system. If no failover target system is available, Parent remains offline.

Table 10-4 Service group dependency configurations: Failover parent / Failover child (*continued*)

Link	Failover parent depends on ...	Failover parent is online If ...	If failover child faults, then ...	If failover parent faults, then ...
online remote soft	Failover Child online on another system in the cluster.	Child is online on another system in the cluster.	<p>If Child fails over to the system on which Parent was online, Parent migrates to another system.</p> <p>If Child fails over to another system, Parent continues to run on original system.</p> <p>If Child cannot fail over, Parent remains online.</p>	<p>Child stays online.</p> <p>Parent fails over to a system where Child is not online.</p> <p>If the only system available is where Child is online, Parent is not brought online.</p> <p>If no failover target system is available, Child remains online.</p>
online remote firm	Failover Child online on another system in the cluster.	Child is online on another system in the cluster.	<p>If Child fails over to the system on which Parent was online, Parent switches to another system.</p> <p>If Child fails over to another system, Parent restarts on original system.</p> <p>If Child cannot fail over, VCS takes the parent offline.</p>	<p>Parent fails over to a system where Child is not online.</p> <p>If the only system available is where Child is online, Parent is not brought online.</p> <p>If no failover target system is available, Child remains online.</p>

Table 10-4 Service group dependency configurations: Failover parent / Failover child (*continued*)

Link	Failover parent depends on ...	Failover parent is online If ...	If failover child faults, then ...	If failover parent faults, then ...
online site soft	Failover Child online on same site.	Child is online in the same site.	<p>Parent stays online.</p> <p>If another Child instance is online or Child fails over to a system within the same site, Parent stays online.</p> <p>If Child fails over to a system in another site, parent migrates to another site where Child is online and depends on Child instance(s) in that site.</p> <p>If Child cannot fail over, Parent remains online.</p>	<p>Child remains online.</p> <p>Parent fails over to another system in the same site maintaining dependency on Child instances in the same site.</p> <p>If Parent cannot failover to a system within the same site, then Parent fails over to a system in another site where at least one instance of child is online in the same site.</p>

Table 10-4 Service group dependency configurations: Failover parent / Failover child (*continued*)

Link	Failover parent depends on ...	Failover parent is online If ...	If failover child faults, then ...	If failover parent faults, then ...
online site firm	Failover Child online in the same site	Child is online in the same site.	<p>Parent taken offline.</p> <p>If another instance of child is online in the same site or child fails over to another system in the same site, Parent migrates to a system in the same site.</p> <p>If no Child instance is online or Child cannot fail over, Parent remains offline.</p>	<p>Child remains online.</p> <p>Parent fails over to another system in same site maintaining dependence on Child instances in the same site.</p> <p>If Parent cannot failover to a system within same site, then Parent fails over to a system in another site where at least one instance of child is online.</p>

Table 10-4 Service group dependency configurations: Failover parent / Failover child (*continued*)

Link	Failover parent depends on ...	Failover parent is online If ...	If failover child faults, then ...	If failover parent faults, then ...
offline local	Failover Child offline on the same system	Child is offline on the same system.	<p>If Child fails over to the system on which parent is not running, parent continues running.</p> <p>If child fails over to system on which parent is running, parent switches to another system, if available.</p> <p>If no failover target system is available for Child to fail over to, Parent continues running.</p>	<p>Parent fails over to system on which Child is not online.</p> <p>If no failover target system is available, Child remains online.</p>

About failover parent / parallel child

With a failover parent and parallel child, no hard dependencies are supported.

[Table 10-5](#) shows service group dependency configurations for Failover parent / Parallel child.

Table 10-5 Service group dependency configurations: Failover parent / Parallel child

Link	Failover parent depends on ...	Failover parent is online if ...	If parallel child faults on a system, then ...	If failover parent faults, then ...
online local soft	Instance of parallel Child group on same system.	Instance of Child is online on same system.	If Child instance fails over to another system, the Parent also fails over to the same system. If Child instance cannot failover to another system, Parent remains online.	Parent fails over to other system and depends on Child instance there. Child Instance remains online where the Parent faulted.
online local firm	Instance of parallel Child group on same system.	Instance of Child is online on same system.	Parent is taken offline. Parent fails over to other system and depends on Child instance there.	Parent fails over to other system and depends on Child instance there. Child Instance remains online where Parent faulted.
online global soft	All instances of parallel Child group online in the cluster.	At least one instance of Child group is online somewhere in the cluster.	Parent remains online if Child faults on any system. If Child cannot fail over to another system, Parent remains online.	Parent fails over to another system, maintaining dependence on all Child instances.

Table 10-5 Service group dependency configurations: Failover parent / Parallel child (*continued*)

Link	Failover parent depends on ...	Failover parent is online if ...	If parallel child faults on a system, then ...	If failover parent faults, then ...
online global firm	One or more instances of parallel Child group remaining online	An instance of Child group is online somewhere in the cluster.	Parent is taken offline. If another Child instance is online or Child fails over, Parent fails over to another system or the same system. If no Child instance is online or Child cannot fail over, Parent remains offline.	Parent fails over to another system, maintaining dependence on all Child instances.
online remote soft	One or more instances parallel Child group remaining online on other systems.	One or more instances of Child group are online on other systems.	Parent remains online. If Child fails over to the system on which Parent is online, Parent fails over to another system.	Parent fails over to another system, maintaining dependence on the Child instances.

Table 10-5 Service group dependency configurations: Failover parent / Parallel child (*continued*)

Link	Failover parent depends on ...	Failover parent is online if ...	If parallel child faults on a system, then ...	If failover parent faults, then ...
online remote firm	All instances parallel Child group remaining online on other systems.	All instances of Child group are online on other systems.	<p>Parent is taken offline.</p> <p>If Child fails over to the system on which Parent is online, Parent fails over to another system.</p> <p>If Child fails over to another system, Parent is brought online on its original system.</p>	Parent fails over to another system, maintaining dependence on all Child instances.
online site soft	One or more instances of parallel Child group in the same site.	At least one instance of Child is online in same site.	<p>Parent stays online if child is online on any system in the same site.</p> <p>If Child fails over to a system in another site, Parent stays in the same site.</p> <p>If Child cannot fail over, Parent remains online.</p>	<p>Child stays online.</p> <p>Parents fails over to a system with Child online in the same site.</p> <p>If the parent group cannot failover, child group remains online</p>

Table 10-5 Service group dependency configurations: Failover parent / Parallel child (*continued*)

Link	Failover parent depends on ...	Failover parent is online if ...	If parallel child faults on a system, then ...	If failover parent faults, then ...
online site firm	One or more instances of parallel Child group in the same site.	At least one instance of Child is online in same site.	Parent stays online if any instance of child is online in the same site. If Child fails over to another system, Parent migrates to a system in the same site. If Child cannot fail over, Parent remains offline.	Child stays online. Parents fails over to a system with Child online in the same site. If the parent group cannot failover, child group remains online.
offline local	Parallel Child offline on same system.	No instance of Child is online on same system.	Parent remains online if Child fails over to another system. If Child fails over to the system on which Parent is online, Parent fails over.	Child remains online and parent fails over to another system where child is not online. If the parent group cannot failover, child group remains offline.

About parallel parent / failover child

Table 10-6 shows service group dependencies for parallel parent / failover child.

Online local dependencies between parallel parent groups and failover child groups are not supported.

Table 10-6 Service group dependency configurations: Parallel parent / Failover child

Link	Parallel parent instances depend on ...	Parallel parent instances are online if ...	If failover child faults on a system, then ...	If parallel parent faults, then ...
online global soft	Failover Child group online somewhere in the cluster.	Failover Child is online somewhere in the cluster.	Parent remains online.	Child remains online
online global firm	Failover Child group somewhere in the cluster.	Failover Child is online somewhere in the cluster.	All instances of Parent taken offline. After Child fails over, Parent instances are failed over or restarted on the same systems.	Child stays online.
online remote soft	Failover Child group on another system.	Failover Child is online on another system.	If Child fails over to system on which Parent is online, Parent fails over to other systems. If Child fails over to another system, Parent remains online.	Child remains online. Parent tries to fail over to another system where child is not online.

Table 10-6 Service group dependency configurations: Parallel parent / Failover child (*continued*)

Link	Parallel parent instances depend on ...	Parallel parent instances are online if ...	If failover child faults on a system, then ...	If parallel parent faults, then ...
online remote firm	Failover Child group on another system.	Failover Child is online on another system.	<p>All instances of Parent taken offline.</p> <p>If Child fails over to system on which Parent was online, Parent fails over to other systems.</p> <p>If Child fails over to another system, Parent brought online on same systems.</p>	Child remains online. Parent tries to fail over to another system where child is not online.
offline local	Failover Child offline on same system.	Failover Child is not online on same system.	<p>Parent remains online if Child fails over to another system.</p> <p>Child fails over to another system. If Parent is online on that system, Parent is brought offline. Parent fails over to any other system.</p>	Child remains online.

About parallel parent / parallel child

Global and remote dependencies between parallel parent groups and parallel child groups are not supported.

Table 10-7 shows service group dependency configurations for parallel parent / parallel child.

Table 10-7 Service group dependency configurations: Parallel parent / Parallel child

Link	Parallel parent depends on ...	Parallel parent is online If ...	If parallel child faults, then ...	If parallel parent faults, then ...
online local soft	Parallel Child instance online on same system.	Parallel Child instance is online on same system.	If Child fails over to another system, Parent migrates to the same system as the Child. If Child cannot fail over, Parent remains online.	Child instance stays online. Parent instance can fail over only to system where Child instance is running and other instance of Parent is not running.
online local firm	Parallel Child instance online on same system.	Parallel Child instance is online on same system.	Parent taken offline. If Child fails over to another system, VCS brings an instance of the Parent online on the same system as Child. If Child cannot fail over, Parent remains offline.	Child stays online. Parent instance can fail over only to system where Child instance is running and other instance of Parent is not running.

Table 10-7 Service group dependency configurations: Parallel parent / Parallel child (*continued*)

Link	Parallel parent depends on ...	Parallel parent is online If ...	If parallel child faults, then ...	If parallel parent faults, then ...
offline local	Parallel Child offline on same system.	No instance of Child is online on same system.	Parent remains online if Child fails over to another system. Parent goes offline if Child fails over to a system where Parent is online. Parent fails over to another system where Child is not online.	Child remains online. Parent fails over to a system where Child is not online.

Frequently asked questions about group dependencies

[Table 10-8](#) lists some commonly asked questions about group dependencies.

Table 10-8 Frequently asked questions about group dependencies

Dependency	Frequently asked questions
Online local	<p>Can child group be taken offline when parent group is online? Soft=Yes Firm=No Hard = No.</p> <p>Can parent group be switched while child group is online? Soft=No Firm=No Hard = No.</p> <p>Can child group be switched while parent group is online? Soft=No Firm=No Hard = No.</p>

Table 10-8 Frequently asked questions about group dependencies
(continued)

Dependency	Frequently asked questions
Online global	<p>Can child group be taken offline when parent group is online? Soft=Yes Firm=No.</p> <p>Can parent group be switched while child group is running? Soft=Yes Firm=Yes.</p> <p>Can child group be switched while parent group is running? Soft=Yes Firm=No</p>
Online remote	<p>Can child group be taken offline when parent group is online? Soft=Yes Firm=No.</p> <p>Can parent group be switched while child group is running? Soft=Yes, but not to system on which child is running. Firm=Yes, but not to system on which child is running.</p> <p>Can child group be switched while parent group is running? Soft=Yes Firm=No, but not to system on which parent is running.</p>
Offline local	<p>Can parent group be brought online when child group is offline? Yes.</p> <p>Can child group be taken offline when parent group is online? Yes.</p> <p>Can parent group be switched while the child group is running? Yes, but not to system on which child is running.</p> <p>Can child group be switched while the parent group is running? Yes, but not to system on which parent is running.</p>
Online site	<p>Can child group be taken offline when parent group is online? Soft=Yes Firm=No.</p> <p>Can parent group be switched to a system in the same site while child group is running? Soft=Yes Firm=Yes.</p> <p>Can child group be switched while parent group is running? Soft=Yes Firm=No</p>

About linking service groups

Note that 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.

Use the following command to link service groups from the command line

```
hagrp -link parent_group child_group gd_category gd_location [gd_type]
```

<i>parent_group</i>	Name of the parent group
<i>child_group</i>	Name of the child group
<i>gd_category</i>	category of group dependency (online/offline).
<i>gd_location</i>	the scope of dependency (local/global/remote/site).
<i>gd_type</i>	type of group dependency (soft/firm/hard). Default is firm

VCS behavior with service group dependencies

VCS enables or restricts service group operations to honor service group dependencies. VCS rejects operations if the operation violates a group dependency.

Online operations in group dependencies

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.

Offline operations in group dependencies

VCS rejects offline operations if the procedure violates existing group dependencies. Typically, firm dependencies are more restrictive to taking child group offline while parent group is online. Rules for manual offline include:

- Parent group offline is never rejected.
- For all soft dependencies, child group can go offline regardless of the state of parent group.
- For all firm dependencies, if parent group is online, child group offline is rejected.
- For the online local hard dependency, if parent group is online, child group offline is rejected.

Switch operations in group dependencies

Switching a service group implies manually taking a service group offline on one system, and manually bringing it back online on another system. VCS rejects manual switch if the group does not comply with the rules for offline or online operations.

VCS event notification

This chapter includes the following topics:

- [About VCS event notification](#)
- [Components of VCS event notification](#)
- [About VCS events and traps](#)
- [About monitoring aggregate events](#)
- [About configuring notification](#)

About VCS event notification

VCS provides a method for notifying important events such as resource or system faults to administrators or designated recipients. VCS includes a notifier component, which consists of the notifier process and the hanotify utility.

VCS supports SNMP consoles that can use an SNMP V2 MIB.

The notifier process performs the following tasks:

- Receives notifications from HAD
- Formats the notification
- Generates an SNMP (V2) trap or sends an email to the designated recipient, or does both.

If you have configured owners for resources, groups, or for the cluster, VCS also notifies owners of the events that affect their resources. A resource owner is notified of resource-related events, a group owner of group-related events, and so on.

You can also configure persons other than owners as recipients of notifications about events of a resource, resource type, service group, system, or cluster. The registered recipients get notifications for the events that have a severity level that

is equal to or greater than the level specified. For example, if you configure recipients for notifications and specify the severity level as Warning, VCS notifies the recipients about events with the severity levels Warning, Error, and SevereError but not about events with the severity level Information.

See [“About attributes and their definitions”](#) on page 524.

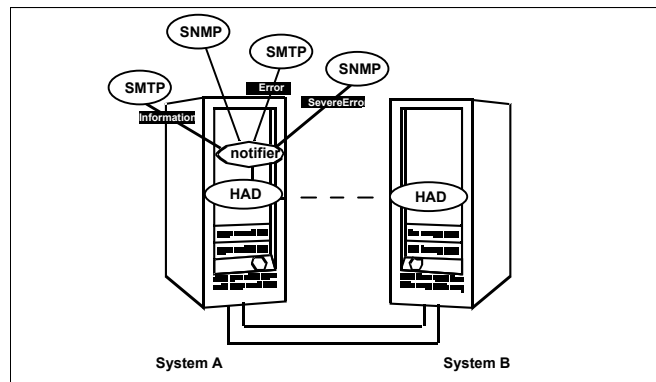
[Figure 11-1](#) shows the severity levels of VCS events.

Table 11-1 VCS event severity levels

Severity level	Denotes
SevereError	Critical errors that can lead to data loss or corruption; SevereError is the highest severity level.
Error	Faults
Warning	Deviations from normal behavior
Information	Important events that exhibit normal behavior; Information is the lowest severity level.

Note: Severity levels are case-sensitive.

Figure 11-1 VCS event notification: Severity levels



SNMP traps are forwarded to the SNMP console. Typically, traps are predefined for events such as service group or resource faults. You can use the hanotify utility to send additional traps.

Event messages and severity levels

When the VCS engine starts up, it queues all messages of severity Information and higher for later processing.

When notifier connects, it communicates to HAD the lowest severity threshold level currently defined for the SNMP option or for the SMTP option.

If notifier is started from the command line without specifying a severity level for the SNMP console or SMTP recipients, notifier communicates the default severity level Warning to HAD. If notifier is configured under VCS control, severity must be specified.

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 labelled severity level Warning and greater.

Notifier ensures that recipients receive only the messages they are 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 behavior prevents messages from being lost between the time HAD stores them and notifier communicates the specifications to HAD.

About persistent and replicated message queue

VCS includes a sophisticated mechanism for maintaining event messages, which ensures that messages are not lost. On each node, VCS queues messages to be sent to the notifier process. This queue is persistent as long as VCS is running and the contents of this queue remain the same on each node. If the notifier service group fails, notifier is failed over to another node in the cluster. Because the message queue is consistent across nodes, notifier can resume message delivery from where it left off even after failover.

How HAD deletes messages

The VCS engine, HAD, stores messages to be sent to notifier. After every 180 seconds, HAD tries to send all the pending notifications to notifier. When HAD receives an acknowledgement from notifier that a message is delivered to at least one of the recipients, it deletes the message from its queue. For example, if two SNMP consoles and two email recipients are designated, notifier sends an

acknowledgement to HAD even if the message reached only one of the four recipients. If HAD does not get acknowledgement for some messages, it keeps on sending these notifications to notifier after every 180 seconds till it gets an acknowledgement of delivery from notifier. An error message is printed to the log file when a delivery error occurs.

HAD deletes messages under the following conditions too:

- The message has been in the queue for time (in seconds) specified in MessageExpiryInterval attribute (default value: one hour) and notifier is unable to deliver the message to the recipient.
- The message queue is full and to make room for the latest message, the earliest message is deleted.

Components of VCS event notification

This topic describes the notifier process and the hanotify utility.

About 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. You can change this value by modifying the MessageQueue attribute.

When notifier is started from the command line, VCS does not control the notifier process. For best results, use the NotifierMngr agent that is bundled with VCS. Configure notifier as part of a highly available service group, which can then be monitored, brought online, and taken offline.

Note that notifier must be configured in a failover group, not parallel, because only one instance of notifier runs in the entire cluster. Also note that notifier does not respond to SNMP `get` or `set` requests; notifier is a trap generator only.

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 that are sent to each.

Note: If you start the notifier outside of VCS control, use the absolute path of the notifier in the command. VCS cannot monitor the notifier process if it is started outside of VCS control using a relative path.

Example of notifier command

Following is an example of a `notifier` command:

```
/opt/VRTSvcs/bin/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:

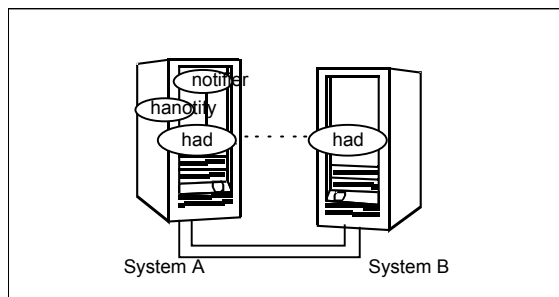
- Sends all level SNMP traps to north at the default SNMP port and community value public.
- Sends Warning traps to north.
- Sends Error and SevereError traps to south at port 2000 and community value your_company.
- Sends SevereError email messages to north as SMTP server at default port and to email recipient abc@your_company.com.

About the hanotify utility

The hanotify utility enables you to construct user-defined messages. The utility forwards messages to HAD, which 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.

Figure 11-2 shows the hanotify utility.

Figure 11-2 hanotify utility



Example of hanotify command

Following is an example of `hanotify` command:

```
hanotify -i 1.3.6.1.4.1.1302.3.8.10.2.8.0.10 -l Warning -n  
agentres -T 7 -t "custom agent" -o 4 -S sys1 -L mv -p  
sys2 -P mv -c MyAgent -C 7 -O johndoe -m "Custom message"
```

In this example, the number 1.3.6.1.4.1.1302.3.8.10.2.8.0.10 is the OID (Object Identifier) for the message being sent. Because it is a user-defined message, HAD has no way of knowing the OID associated with the SNMP trap corresponding to this message. Users must provide the OID.

The message severity level is set to Warning. The affected systems are sys1 and sys2. Running this command generates a custom notification with the message "Custom message" for the resource *agentres*.

About VCS events and traps

This topic lists the events that generate traps, email notification, or both. Note that SevereError indicates the highest severity level, Information the lowest. Traps specific to global clusters are ranked from Critical, the highest severity, to Normal, the lowest.

Events and traps for clusters

[Table 11-2](#) shows events and traps for clusters.

Table 11-2 Events and traps for clusters

Event	Severity level	Description
Cluster has faulted.	Error	The cluster is down because of a fault.
Heartbeat is down. (Global Cluster Option)	Error	The connector on the local cluster lost its heartbeat connection to the remote cluster.
Remote cluster is in RUNNING state. (Global Cluster Option)	Information	Local cluster has complete snapshot of the remote cluster, indicating the remote cluster is in the RUNNING state.
Heartbeat is "alive." (Global Cluster Option)	Information	The heartbeat between clusters is healthy.

Table 11-2 Events and traps for clusters (*continued*)

Event	Severity level	Description
User has logged on to VCS.	Information	A user log on has been recognized because a user logged on by Cluster Manager, or because a <code>haxxx</code> command was invoked.

Events and traps for agents

[Table 11-3](#) depicts events and traps for agents.

Table 11-3 Events and traps for agents

Event	Severity level	Description
Agent is faulted.	Warning	The agent has faulted on one node in the cluster.
Agent is restarting	Information	VCS is restarting the agent.

Events and traps for resources

[Table 11-4](#) depicts events and traps for resources.

Table 11-4 Events and traps for resources

Event	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.
Health of cluster resource declined.	Warning	Used by agents to give additional information on the state of a resource. A decline in the health of the resource was identified during monitoring.

Table 11-4 Events and traps for resources (*continued*)

Event	Severity level	Description
Resource went online by itself.	Warning (not for first probe)	The resource was brought online on its own.
Resource has faulted.	Error	The resource has faulted on one node in the cluster.
Resource is being restarted by agent.	Information	The agent is restarting the resource.
The health of cluster resource improved.	Information	Used by agents to give extra information about state of resource. An improvement in the health of the resource was identified during monitoring.
Resource monitor time has changed.	Warning	<p>This trap is generated when statistical analysis for the time taken by the monitor function of an agent is enabled for the agent.</p> <p>See “VCS agent statistics” on page 464.</p> <p>This trap is generated when the agent framework detects a sudden change in the time taken to run the monitor function for a resource. The trap information contains details of:</p> <ul style="list-style-type: none">■ The change in time required to run the monitor function■ The actual times that were compared to deduce this change.
Resource is in ADMIN_WAIT state.	Error	<p>The resource is in the admin_wait state.</p> <p>See “About controlling Clean behavior on resource faults” on page 297.</p>

Events and traps for systems

Table 11-5 depicts events and traps for systems.

Table 11-5 Events and traps for systems

Event	Severity level	Description
VCS is being restarted by <code>hashadow</code> .	Warning	The <code>hashadow</code> process is restarting the VCS engine.
VCS is in jeopardy.	Warning	One node running VCS is in jeopardy.
VCS is up on the first node in the cluster.	Information	VCS is up on the first node.
VCS has faulted.	SevereError	VCS is down because of a fault.
A node running VCS has joined cluster.	Information	The cluster has a new node that runs VCS.
VCS has exited manually.	Information	VCS has exited gracefully from one node on which it was previously running.
CPU usage exceeded threshold on the system.	Warning	The system's CPU usage exceeded the Warning threshold level set in the <code>CPUThreshold</code> attribute.
Swap usage exceeded threshold on the system.	Warning	The system's swap usage exceeded the Warning threshold level set in the <code>SwapThreshold</code> attribute.
Memory usage exceeded threshold on the system.	Warning	The system's Memory usage exceeded the Warning threshold level set in the <code>MemThresholdLevel</code> attribute.

Events and traps for service groups

Table 11-6 depicts events and traps for service groups.

Table 11-6 Events and traps for service groups

Event	Severity level	Description
Service group has faulted.	Error	The service group is offline because of a fault.
Service group concurrency violation.	SevereError	A failover service group has become online on more than one node in the cluster.
Service group has faulted and cannot be failed over anywhere.	SevereError	Specified service group faulted on all nodes where group could be brought online. There are no nodes to which the group can fail over.
Service group is online	Information	The service group is online.
Service group is offline.	Information	The service group is offline.
Service group is autodisabled.	Information	VCS has autodisabled the specified group because one node exited the cluster.
Service group is restarting.	Information	The service group is restarting.
Service group is being switched.	Information	VCS is taking the service group offline on one node and bringing it online on another.
Service group restarting in response to persistent resource going online.	Information	The service group is restarting because a persistent resource recovered from a fault.
The global service group is online/partial on multiple clusters. (Global Cluster Option)	SevereError	A concurrency violation occurred for the global service group.
Attributes for global service groups are mismatched. (Global Cluster Option)	Error	The attributes ClusterList, AutoFailOver, and Parallel are mismatched for the same global service group on different clusters.

SNMP-specific files

VCS includes two SNMP-specific files: `vcs.mib` and `vcs_trapd`, which are created in:

`%VCS_HOME%\snmp.`

The file `vcs.mib` is the textual MIB for built-in traps that are supported by VCS. Load this MIB into your 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. The file includes sample events configured for the built-in SNMP traps supported by VCS. To merge these events with those configured for SNMP traps:

```
xnmevents -merge vcs_trapd
```

When you merge events, the SNMP traps sent by VCS by way of notifier are displayed in the HP OpenView NNM SNMP console.

Note: For more information on `xnmevents`, see the HP OpenView documentation.

Trap variables in VCS MIB

Traps sent by VCS are reversible to SNMPv2 after an SNMPv2 to SNMPv1 conversion.

For reversible translations between SNMPv1 and SNMPv2 trap PDUs, the second-last ID of the SNMP trap OID must be zero. This ensures that once you make a forward translation (SNMPv2 trap to SNMPv1; RFC 2576 Section 3.2), the reverse translation (SNMPv1 trap to SNMPv2 trap; RFC 2576 Section 3.1) is accurate.

The VCS notifier follows this guideline by using OIDs with second-last ID as zero, enabling reversible translations.

About severityId

This variable indicates the severity of the trap being sent.

[Table 11-7](#) shows the values that the variable `severityId` can take.

Table 11-7 Possible values of the variable severityId

Severity level and description	Value in trap PDU
Information Important events exhibiting normal behavior	0
Warning Deviation from normal behavior	1
Error A fault	2
Severe Error Critical error that can lead to data loss or corruption	3

EntityType and entitySubType

These variables specify additional information about the entity.

[Table 11-8](#) shows the variables entityType and entitySubType.

Table 11-8 Variables entityType and entitySubType

Entity type	Entity sub-type
Resource	String. For example, disk.
Group	String The type of the group (failover or parallel)
System	
Heartbeat	String Type of the heartbeat
VCS	String
GCO	String
Agent name	String The agent name

About entityState

This variable describes the state of the entity.

Table 11-9 shows the the various states.

Table 11-9 Possible states

Entity	States
VCS states	<ul style="list-style-type: none">■ User has logged into VCS■ Cluster has faulted■ Cluster is in RUNNING state
Agent states	<ul style="list-style-type: none">■ Agent is restarting■ Agent has faulted
Resources states	<ul style="list-style-type: none">■ Resource state is unknown■ Resource monitoring has timed out■ Resource is not going offline■ Resource is being restarted by agent■ Resource went online by itself■ Resource has faulted■ Resource is in admin wait state■ Resource monitor time has changed
Service group states	<ul style="list-style-type: none">■ Service group is online■ Service group is offline■ Service group is auto disabled■ Service group has faulted■ Service group has faulted and cannot be failed over anywhere■ Service group is restarting■ Service group is being switched■ Service group concurrency violation■ Service group is restarting in response to persistent resource going online■ Service group attribute value does not match corresponding remote group attribute value■ Global group concurrency violation

Table 11-9 Possible states (*continued*)

Entity	States
System states	<ul style="list-style-type: none">■ VCS is up on the first node in the Cluster■ VCS is being restarted by hashadow■ VCS is in jeopardy■ VCS has faulted■ A node running VCS has joined cluster■ VCS has exited manually■ CPU usage exceeded the threshold on the system■ Memory usage exceeded the threshold on the system■ Swap usage exceeded the threshold on the system
GCO heartbeat states	<ul style="list-style-type: none">■ Cluster has lost heartbeat with remote cluster■ Heartbeat with remote cluster is alive

About monitoring aggregate events

This topic describes how you can detect aggregate events by monitoring individual notifications.

How to detect service group failover

VCS does not send any explicit traps when a failover occurs in response to a service group fault. When a service group faults, VCS generates the following notifications if the AutoFailOver attribute for the service group is set to 1:

- Service Group Fault for the node on which the service group was online and faulted
- Service Group Offline for the node on which the service group faulted
- Service Group Online for the node to which the service group failed over

How to detect service group switch

When a service group is switched, VCS sends a notification of severity Information to indicate the following events:

- Service group is being switched.
- Service group is offline for the node from which the service group is switched.
- Service group is online for the node to which the service group was switched. This notification is sent after VCS completes the service group switch operation.

Note: You must configure appropriate severity for the notifier to receive these notifications. To receive VCS notifications, the minimum acceptable severity level is Information.

About configuring notification

Configuring notification involves creating a resource for the Notifier Manager (NotifierMgr) agent in the ClusterService group.

VCS provides several methods for configuring notification:

- Manually editing the main.cf file.
- Using the Notifier wizard.

VCS event triggers

This chapter includes the following topics:

- [About VCS event triggers](#)
- [Using event triggers](#)
- [List of event triggers](#)

About VCS event triggers

Triggers let you invoke user-defined scripts for specified events in a cluster.

VCS determines if the event is enabled and invokes the `hatrigger` script. The script is located at:

```
%VCS_HOME%\bin\hatrigger.pl
```

VCS also passes the name of the event trigger and associated parameters. For example, when a service group comes online on a system, VCS invokes the following command:

```
hatrigger -postonline system service_group
```

VCS does not wait for the trigger to complete execution. VCS calls the trigger and continues normal operation.

VCS invokes event triggers on the system where the event occurred, with the following exceptions:

- VCS invokes the `sysoffline` and `nofailover` event triggers on the lowest-numbered system in the `RUNNING` state.
- VCS invokes the `violation` event trigger on all systems on which the service group was brought partially or fully online.

By default, the `hattrigger` script invokes the trigger script(s) from the default path `$VCS_HOME/bin/triggers`. You can customize the trigger path by using the `TriggerPath` attribute.

See [“Resource attributes”](#) on page 525.

See [“Service group attributes”](#) on page 546.

The same path is used on all nodes in the cluster. The trigger path must exist on all the cluster nodes. On each cluster node, the trigger scripts must be installed in the trigger path.

Using event triggers

VCS provides a sample Perl script for each event trigger at the following location:

`%VCS_HOME%\bin\sample_triggers`

Customize the scripts according to your requirements: you may choose to write your own Perl scripts.

To use an event trigger

- 1 Use the sample scripts to write your own custom actions for the trigger.
- 2 Move the modified trigger script to the following path on each node:
`%VCS_HOME%\bin\triggers`
- 3 Configure other attributes that may be required to enable the trigger. See the usage information for the trigger for more information.

List of event triggers

The information in the following sections describes the various event triggers, including their usage, parameters, and location.

About the dumptunables trigger

The following table describes the dumptunables event trigger:

Description	<p>The dumptunables trigger is invoked when HAD goes into the RUNNING state. When this trigger is invoked, it uses the HAD environment variables that it inherited, and other environment variables to process the event. Depending on the value of the <i>to_log</i> parameter, the trigger then redirects the environment variables to either stdout or the engine log.</p> <p>This trigger is not invoked when HAD is restarted by hashadow.</p> <p>This event trigger is internal and non-configurable.</p>
Usage	<p><code>-dumptunables <i>triggertype</i> <i>system</i> <i>to_log</i></code></p> <p><i>triggertype</i>—represents whether trigger is custom (<i>triggertype</i>=0) or internal (<i>triggertype</i>=1).</p> <p>For this trigger, <i>triggertype</i>=0.</p> <p><i>system</i>—represents the name of the system on which the trigger is invoked.</p> <p><i>to_log</i>—represents whether the output is redirected to engine log (<i>to_log</i>=1) or stdout (<i>to_log</i>=0).</p>

About the injeopardy event trigger

The following table describes the injeopardy event trigger:

Description	<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 event is considered critical because if the system loses the remaining network link, VCS does not fail over the service groups that were online on the system. Use this trigger to notify the administrator of the critical event. The administrator can then 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>
Usage	<p><code>-injeopardy <i>triggertype</i> <i>system</i> <i>system_state</i></code></p> <p><i>triggertype</i>—represents whether trigger is custom (<i>triggertype</i>=0) or internal (<i>triggertype</i>=1).</p> <p>For this trigger, <i>triggertype</i>=0.</p> <p><i>system</i>—represents the name of the system.</p> <p><i>system_state</i>—represents the value of the State attribute.</p>

About the loadwarning event trigger

The following table describes the loadwarning event trigger:

Description	<p>Invoked when a system becomes overloaded because the load of the system's online groups exceeds the system's LoadWarningLevel attribute for an interval exceeding the LoadTimeThreshold attribute.</p> <p>For example, assume that the Capacity 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 $0.80 \times 150 = 120$. The trigger is invoked if the system load stays at 135 for more than 300 seconds because the actual load is above the limit of 120 specified by LoadWarningLevel.</p> <p>Use this trigger to notify the administrator of the critical event. The administrator can then switch some service groups to another system, ensuring that no one system is overloaded.</p> <p>This event trigger is non-configurable.</p>
Usage	<p><code>-loadwarning <i>triggertype</i> <i>system</i> <i>available_capacity</i></code></p> <p><i>triggertype</i>—represents whether trigger is custom (<i>triggertype</i>=0) or internal (<i>triggertype</i>=1).</p> <p>For this trigger, <i>triggertype</i>=0.</p> <p><i>system</i>—represents the name of the system.</p> <p><i>available_capacity</i>—represents the system's AvailableCapacity attribute. (AvailableCapacity=Capacity-sum of Load for system's online groups.)</p>

About the nofailover event trigger

The following table describes the nofailover event trigger:

Description	<p>Called from the lowest-numbered system in RUNNING state when a service group cannot fail over.</p> <p>This event trigger is non-configurable.</p>
Usage	<p><code>-nofailover <i>triggertype</i> <i>system</i> <i>service_group</i></code></p> <p><i>triggertype</i>—represents whether trigger is custom (<i>triggertype</i>=0) or internal (<i>triggertype</i>=1).</p> <p>For this trigger, <i>triggertype</i>=0.</p> <p><i>system</i>—represents the name of the last system on which an attempt was made to bring the service group online.</p> <p><i>service_group</i>—represents the name of the service group.</p>

About the postoffline event trigger

The following table describes the postoffline event trigger:

Description	<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>
Usage	<p><code>-postoffline <i>triggertype</i> <i>system</i> <i>service_group</i></code></p> <p><i>triggertype</i>—represents whether trigger is custom (<i>triggertype</i>=0) or internal (<i>triggertype</i>=1).</p> <p>For this trigger, <i>triggertype</i>=0.</p> <p><i>system</i>—represents the name of the system.</p> <p><i>service_group</i>—represents the name of the service group that went offline.</p>

About the postonline event trigger

The following table describes the postonline event trigger:

Description	<p>This event trigger is invoked on the system where the group went online from an offline state.</p> <p>This event trigger is non-configurable.</p>
Usage	<p><code>-postonline <i>triggertype</i> <i>system</i> <i>service_group</i></code></p> <p><i>triggertype</i>—represents whether trigger is custom (<i>triggertype</i>=0) or internal (<i>triggertype</i>=1).</p> <p>For this trigger, <i>triggertype</i>=0.</p> <p><i>system</i>—represents the name of the system.</p> <p><i>service_group</i>—represents the name of the service group that went online.</p>

About the preonline event trigger

The following table describes the preonline event trigger:

Description	<p>Indicates when the HAD should call a user-defined script before bringing a service group online in response to the <code>hagrp -online</code> command or a fault.</p> <p>If the trigger does not exist, VCS continues to bring the group online. If the script returns 0 without an exit code, VCS runs the <code>hagrp -online -nopre</code> command, with the <code>-checkpartial</code> option if appropriate.</p> <p>If you do want to bring the group online, define the trigger to take no action. This event trigger is configurable.</p>
Usage	<pre>-preonline <i>triggertype</i> <i>system</i> <i>service_group</i> <i>whyonlining</i> [<i>system_where_group_faulted</i>]</pre> <p><i>triggertype</i>—represents whether trigger is custom (<i>triggertype</i>=0) or internal (<i>triggertype</i>=1).</p> <p>For this trigger, <i>triggertype</i>=0.</p> <p><i>system</i>—represents the name of the system.</p> <p><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><i>whyonlining</i>—represents two values:</p> <p>FAULT: Indicates that the group was brought online in response to a group failover.</p> <p>MANUAL: Indicates that group was brought online or switched manually on the system that is represented by the variable <i>system</i>.</p> <p><i>system_where_group_faulted</i>—represents the name of the system on which the group has faulted or switched. This variable is optional and set when the engine invokes the trigger during a failover or switch.</p>
To enable the trigger	<p>Set the PreOnline attribute in the service group definition to 1.</p> <p>You can set a local (per-system) value for the attribute to control behavior on each node in the cluster.</p>
To disable the trigger	<p>Set the PreOnline attribute in the service group definition to 0.</p> <p>You can set a local (per-system) value for the attribute to control behavior on each node in the cluster.</p>

About the resadminwait event trigger

The following table describes the resadminwait event trigger:

Description	<p>Invoked when a resource enters ADMIN_WAIT state.</p> <p>When VCS sets a resource in the ADMIN_WAIT state, it invokes the resadminwait trigger according to the reason the resource entered the state.</p> <p>See “Clearing resources in the ADMIN_WAIT state” on page 298.</p> <p>This event trigger is non-configurable.</p>
Usage	<pre>-resadminwait system resource adminwait_reason</pre> <p><i>system</i>—represents the name of the system.</p> <p><i>resource</i>—represents the name of the faulted resource.</p> <p><i>adminwait_reason</i>—represents the reason the resource entered the ADMIN_WAIT state. Values range from 0-5:</p> <p>0 = The offline function did not complete within the expected time.</p> <p>1 = The offline function was ineffective.</p> <p>2 = The online function did not complete within the expected time.</p> <p>3 = The online function was ineffective.</p> <p>4 = The resource was taken offline unexpectedly.</p> <p>5 = The monitor function consistently failed to complete within the expected time.</p>

About the resfault event trigger

The following table describes the resfault event trigger:

Description	<p>Invoked on the system where a resource has 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 configurable.</p> <p>To configure this trigger, you must define the following:</p> <p>TriggerResFault: Set the attribute to 1 to invoke the trigger when a resource faults.</p>
-------------	---

Usage	<div><div><code>-resfault <i>triggertype</i> <i>system</i> <i>resource</i> <i>previous_state</i></code></div><div><i>triggertype</i>—represents whether trigger is custom (<i>triggertype</i>=0) or internal (<i>triggertype</i>=1).</div><div>For this trigger, <i>triggertype</i>=0.</div><div><i>system</i>—represents the name of the system.</div><div><i>resource</i>—represents the name of the faulted resource.</div><div><i>previous_state</i>—represents the resource's previous state.</div></div>
To enable the trigger	<div>To invoke the trigger when a resource faults, set the TriggerResFault attribute to 1.</div>

About the resnotoff event trigger

The following table describes the resnotoff event trigger:

Description	<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.</p> <p>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 <code>shutdown</code> 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:</p> <p>Resource Name 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>\$shutdown_timeout 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>\$shutdown_countdown 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>\$forced_close_app 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>\$reboot_option 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>
Usage	<pre>-resnotoff <i>triggertype system resource</i></pre> <p><i>triggertype</i>—represents whether trigger is custom (<i>triggertype</i>=0) or internal (<i>triggertype</i>=1).</p> <p>For this trigger, <i>triggertype</i>=0.</p> <p><i>system</i>—represents the system on which the resource is not going offline.</p> <p><i>resource</i>—represents the name of the resource.</p>

About the resrestart event trigger

The following table describes the resrestart event trigger.

Description	This trigger is invoked when a resource is restarted by an agent because resource faulted and RestartLimit was greater than 0.
Usage	<div><pre>-resrestart <i>triggertype</i> <i>system</i> <i>resource</i></pre><p><i>triggertype</i>—represents whether trigger is custom (<i>triggertype</i>=0) or internal (<i>triggertype</i>=1).</p><p>For this trigger, <i>triggertype</i>=0.</p><p><i>system</i>—represents the name of the system.</p><p><i>resource</i>—represents the name of the resource.</p></div> <div><p>To enable the trigger</p><p>This event trigger is not enabled by default. You must enable resrestart by setting the attribute TriggerResRestart to 1 in the main.cf file, or by issuing the command:</p><pre>hagrp -modify <i>service_group</i> TriggerResRestart 1</pre><p>However, the attribute is configurable at the resource level. To enable resrestart for a particular resource, you can set the attribute TriggerResRestart to 1 in the main.cf file or issue the command:</p><pre>hares -modify <i>resource</i> TriggerResRestart 1</pre></div>

About the resstatechange event trigger

The following table describes the resstatechange event trigger:

Description	<p>This trigger is invoked under the following conditions:</p> <p>Resource goes from OFFLINE to ONLINE.</p> <p>Resource goes from ONLINE to OFFLINE.</p> <p>Resource goes from ONLINE to FAULTED.</p> <p>Resource goes from FAULTED to OFFLINE. (When fault is cleared on non-persistent resource.)</p> <p>Resource goes from FAULTED to ONLINE. (When faulted persistent resource goes online or faulted non-persistent resource is brought online outside VCS control.)</p> <p>Resource is restarted by an agent because resource faulted and RestartLimit was greater than 0.</p> <p>Warning: In later releases, you cannot use <code>resstatechange</code> to indicate restarting of a resource. Instead, use <code>resrestart</code>. See “About the resrestart event trigger” on page 381.</p> <p>This event trigger is configurable.</p>
Usage	<pre>-resstatechange <i>triggertype</i> <i>system</i> <i>resource</i> <i>previous_state</i> <i>new_state</i></pre> <p><i>triggertype</i>—represents whether trigger is custom (<i>triggertype</i>=0) or internal (<i>triggertype</i>=1).</p> <p>For this trigger, <i>triggertype</i>=0.</p> <p><i>system</i>—represents the name of the system.</p> <p><i>resource</i>—represents the name of the resource.</p> <p><i>previous_state</i>—represents the resource's previous state.</p> <p><i>new_state</i>—represents the resource's new state.</p>

To enable the trigger	<p>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>hagrp -modify service_group TriggerResStateChange 1</pre> <p>Use the <code>resstatechange</code> trigger carefully. For example, enabling this trigger for a service group with 100 resources means that 100 <code>hattrigger</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> <p>However, the attribute is configurable at the resource level. To enable <code>resstatechange</code> for a particular resource, you can set the attribute <code>TriggerResStateChange</code> to 1 in the <code>main.cf</code> file or issue the command:</p> <pre>hares -modify resource TriggerResStateChange 1</pre>
-----------------------	---

About the `sysoffline` event trigger

The following table describes the `sysoffline` event trigger:

Description	<p>Called from the lowest-numbered system in <code>RUNNING</code> state when a system is in jeopardy state or leaves the cluster.</p> <p>This event trigger is non-configurable.</p>
Usage	<pre>-sysoffline system system_state</pre> <p><i>system</i>—represents the name of the system.</p> <p><i>system_state</i>—represents the value of the <code>State</code> attribute.</p> <p>See “System states” on page 521.</p>

About the `unable_to_restart_agent` event trigger

The following table describes the `unable_to_restart_agent` event trigger:

Description	<p>This trigger is invoked when an agent faults more than a predetermined number of times with in an hour. When this occurs, VCS gives up trying to restart the agent. VCS invokes this trigger on the node where the agent faults.</p> <p>You can use this trigger to notify the administrators that an agent has faulted, and that VCS is unable to restart the agent. The administrator can then take corrective action.</p>
-------------	---

Usage	<code>-unable_to_restart_agent system resource_type</code> <i>system</i> —represents the name of the system. <i>resource_type</i> —represents the resource type associated with the agent.
To disable the trigger	Remove the files associated with the trigger from the <code>\$VCS_HOME/bin/triggers</code> directory.

About the `unable_to_restart_had` event trigger

The following table describes the `unable_to_restart_had` event trigger:

Description	<p>This event trigger is invoked by hashadow when hashadow cannot restart HAD on a system. If HAD fails to restart after six attempts, hashadow invokes the trigger on the system.</p> <p>The default behavior of the trigger is to reboot the system. However, service groups previously running on the system are autodisabled when hashadow fails to restart HAD. Before these service groups can be brought online elsewhere in the cluster, you must autoenable them on the system. To do so, customize the <code>unable_to_restart_had</code> trigger to remotely execute the following command from any node in the cluster where VCS is running:</p> <pre>hagrp -autoenable service_group -sys system</pre> <p>For example, if hashadow fails to restart HAD on <i>system1</i>, and if <i>group1</i> and <i>group2</i> were online on that system, a trigger customized in this manner would autoenable <i>group1</i> and <i>group2</i> on <i>system1</i> before rebooting. Autoenabling <i>group1</i> and <i>group2</i> on <i>system1</i> enables these two service groups to come online on another system when the trigger reboots <i>system1</i>.</p> <p>This event trigger is non-configurable.</p>
Usage	<code>-unable_to_restart_had</code> This trigger has no arguments.

About the violation event trigger

The following table describes the violation event trigger:

Description	<p>This trigger is invoked only on the system that caused the concurrency violation. Specifically, it takes the service group offline on the system where the trigger was invoked. Note that this trigger applies to failover groups only. The default trigger takes the service group offline on the system that caused the concurrency violation.</p> <p>This event trigger is internal and non-configurable.</p>
Usage	<pre>-violation <i>system</i> <i>service_group</i></pre> <p><i>system</i>—represents the name of the system.</p> <p><i>service_group</i>—represents the name of the service group that was fully or partially online.</p>

Cluster configurations for disaster recovery

- [Chapter 13. Connecting clusters—Creating global clusters](#)
- [Chapter 14. Administering global clusters from Cluster Manager \(Java console\)](#)
- [Chapter 15. Administering global clusters from the command line](#)
- [Chapter 16. Setting up replicated data clusters](#)

Connecting clusters—Creating global clusters

This chapter includes the following topics:

- [How VCS global clusters work](#)
- [VCS global clusters: The building blocks](#)
- [Prerequisites for global clusters](#)
- [Setting up a global cluster](#)
- [About cluster faults](#)
- [About setting up a disaster recovery fire drill](#)
- [Multi-tiered application support using the RemoteGroup agent in a global environment](#)
- [Test scenario for a multi-tiered environment](#)

How VCS global clusters work

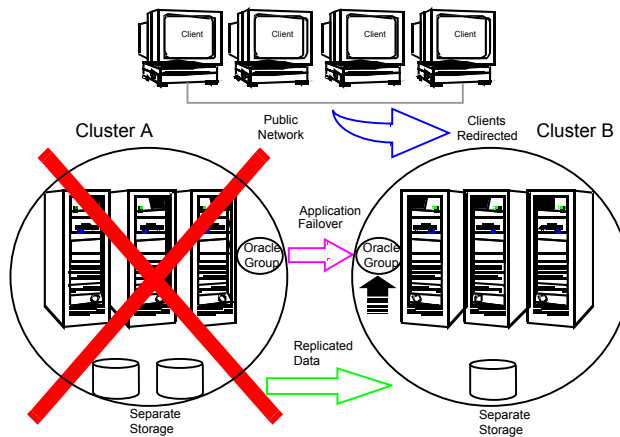
Local clustering provides local failover for each site or building. But, these configurations do not provide protection against large-scale disasters such as major floods, hurricanes, and earthquakes that cause outages for an entire city or region. The entire cluster could be affected by an outage.

In such situations, VCS global clusters ensure data availability by migrating applications to remote clusters located considerable distances apart.

Let us take the example of an Oracle database configured in a VCS global cluster. Oracle is installed and configured in both clusters. Oracle data is located on shared disks within each cluster and is replicated across clusters to ensure data concurrency. The Oracle service group is online on a system in cluster A and is configured to fail over globally, on clusters A and B.

Figure 13-1 shows a sample global cluster setup.

Figure 13-1 Sample global cluster setup



VCS continuously monitors and communicates events between clusters. Inter-cluster communication ensures that the global cluster is aware of the state of the service groups that are configured in the global cluster at all times.

In the event of a system or application failure, VCS fails over the Oracle service group to another system in the same cluster. If the entire cluster fails, VCS fails over the service group to the remote cluster, which is part of the global cluster. VCS also redirects clients once the application is online on the new location.

VCS global clusters: The building blocks

VCS extends clustering concepts to wide-area high availability and disaster recovery with the following:

- Remote cluster objects
See “[Visualization of remote cluster objects](#)” on page 389.
- Global service groups
See “[About global service groups](#)” on page 389.
- Global cluster management

See [“About global cluster management”](#) on page 389.

- Serialization
See [“About serialization—The Authority attribute”](#) on page 391.
- Resiliency and right of way
See [“About resiliency and “Right of way””](#) on page 392.
- VCS agents to manage wide-area failover
See [“VCS agents to manage wide-area failover”](#) on page 392.
- Split-brain in two-cluster global clusters
See [“About the Steward process: Split-brain in two-cluster global clusters”](#) on page 392.
- Secure communication
See [“Secure communication in global clusters”](#) on page 393.

Visualization of remote cluster objects

VCS enables you to visualize remote cluster objects using any of the supported components that are used to administer VCS such as VCS CLI and Veritas InfoScale Operations Manager

See [“Components for administering VCS”](#) on page 36.

You can define remote clusters in your configuration file, `main.cf`. The Remote Cluster Configuration wizard provides an easy interface to do so. The wizard updates the `main.cf` files of all connected clusters with the required configuration changes.

See [“Adding a remote cluster”](#) on page 421.

About global service groups

A global service group is a regular VCS group with additional properties to enable wide-area failover. The global service group attribute `ClusterList` defines the list of clusters to which the group can fail over. The service group must be configured on all participating clusters and must have the same name on each cluster. The Global Group Configuration Wizard provides an easy interface to configure global groups.

See [“Administering global service groups”](#) on page 426.

About global cluster management

VCS enables you to perform operations (online, offline, switch) on global service groups from any system in any cluster. You must log on with adequate privileges for cluster operations.

See [“User privileges for cross-cluster operations”](#) on page 69.

You can bring service groups online or switch them to any system in any cluster. If you do not specify a target system, VCS uses the FailOverPolicy to determine the system.

See [“About defining failover policies”](#) on page 295.

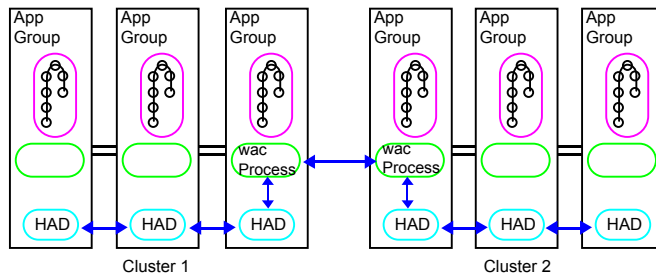
Management of remote cluster objects is aided by inter-cluster communication enabled by the wide-area connector (wac) process.

About the wide-area connector process

The wide-area connector (wac) is a failover Process resource that ensures communication between clusters.

[Figure 13-2](#) is an illustration of the wide-area connector process.

Figure 13-2 Wide-area connector (wac) process



The wac process runs on one system in each cluster and connects with peers in remote clusters. It receives and transmits information about the status of the cluster, service groups, and systems. This communication enables VCS to create a consolidated view of the status of all the clusters configured as part of the global cluster. The process also manages wide-area heartbeating to determine the health of remote clusters. The process also transmits commands between clusters and returns the result to the originating cluster.

VCS provides the option of securing the communication between the wide-area connectors.

See [“Secure communication in global clusters”](#) on page 393.

About the wide-area heartbeat agent

The wide-area heartbeat agent manages the inter-cluster heartbeat. Heartbeats are used to monitor the health of remote clusters. VCS wide-area heartbeat agents include lcmp and lcmpS. While other VCS resource agents report their status to

VCS engine, heartbeat agents report their status directly to the WAC process. The heartbeat name must be the same as the heartbeat type name. You can add only one heartbeat of a specific heartbeat type.

You can create custom wide-area heartbeat agents. For example, the VCS replication agent for SRDF includes a custom heartbeat agent for Symmetrix arrays.

You can add heartbeats using the `hahb -add heartbeatname` command and change the default values of the heartbeat agents using the `hahb -modify` command.

See [“Administering heartbeats in a global cluster setup”](#) on page 445.

See [“Heartbeat attributes \(for global clusters\)”](#) on page 596.

About serialization—The Authority attribute

VCS ensures that global service group operations are conducted serially to avoid timing problems and to ensure smooth performance. The Authority attribute prevents a service group from coming online in multiple clusters at the same time. Authority is a persistent service group attribute and it designates which cluster has the right to bring a global service group online. The attribute cannot be modified at runtime.

If two administrators simultaneously try to bring a service group online in a two-cluster global group, one command is honored, and the other is rejected based on the value of the Authority attribute.

The attribute prevents bringing a service group online in a cluster that does not have the authority to do so. If the cluster holding authority is down, you can enforce a takeover by using the command `hagrp -online -force service_group`. This command enables you to fail over an application to another cluster when a disaster occurs.

Note: A cluster assuming authority for a group does not guarantee the group will be brought online on the cluster. The attribute merely specifies the right to attempt bringing the service group online in the cluster. The presence of Authority does not override group settings like frozen, autodisabled, non-probed, and so on, that prevent service groups from going online.

You must seed authority if it is not held on any cluster.

Offline operations on global groups can originate from any cluster and do not require a change of authority to do so, because taking a group offline does not necessarily indicate an intention to perform a cross-cluster failover.

About the Authority and AutoStart attributes

The attributes Authority and AutoStart work together to avoid potential concurrency violations in multi-cluster configurations.

If the AutoStartList attribute is set, and if a group's Authority attribute is set to 1, the VCS engine waits for the wac process to connect to the peer. If the connection fails, it means the peer is down and the AutoStart process proceeds. If the connection succeeds, HAD waits for the remote snapshot. If the peer is holding the authority for the group and the remote group is online (because of takeover), the local cluster does not bring the group online and relinquishes authority.

If the Authority attribute is set to 0, AutoStart is not invoked.

About resiliency and "Right of way"

VCS global clusters maintain resiliency using the wide-area connector process and the ClusterService group. The wide-area connector process runs as long as there is at least one surviving node in a cluster.

The wide-area connector, its alias, and notifier are components of the ClusterService group.

VCS agents to manage wide-area failover

VCS agents now manage external objects that are part of wide-area failover. These objects include replication, DNS updates, and so on. These agents provide a robust framework for specifying attributes and restarts, and can be brought online upon fail over.

About the Steward process: Split-brain in two-cluster global clusters

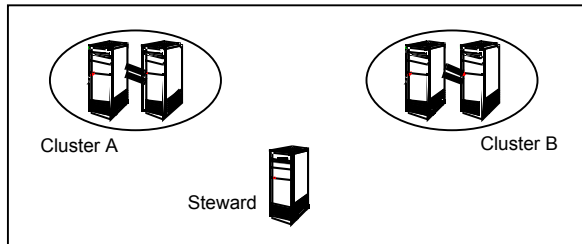
Failure of all heartbeats between any two clusters in a global cluster indicates one of the following:

- The remote cluster is faulted.
- All communication links between the two clusters are broken.

In global clusters with three or more clusters, VCS queries the connected clusters to confirm that the remote cluster is truly down. This mechanism is called inquiry.

In a two-cluster setup, VCS uses the Steward process to minimize chances of a wide-area split-brain. The process runs as a standalone binary on a system outside of the global cluster configuration.

[Figure 13-3](#) depicts the Steward process to minimize chances of a split brain within a two-cluster setup.

Figure 13-3 Steward process: Split-brain in two-cluster global clusters

When all communication links between any two clusters are lost, each cluster contacts the Steward with an inquiry message. The Steward sends an ICMP ping to the cluster in question and responds with a negative inquiry if the cluster is running or with positive inquiry if the cluster is down. The Steward can also be used in configurations with more than two clusters. VCS provides the option of securing communication between the Steward process and the wide-area connectors.

See “[Secure communication in global clusters](#)” on page 393.

In non-secure configurations, you can configure the steward process on a platform that is different to that of the global cluster nodes. Secure configurations have not been tested for running the steward process on a different platform.

A Steward is effective only if there are independent paths from each cluster to the host that runs the Steward. If there is only one path between the two clusters, you must prevent split-brain by confirming manually via telephone or some messaging system with administrators at the remote site if a failure has occurred. By default, VCS global clusters fail over an application across cluster boundaries with administrator confirmation. You can configure automatic failover by setting the `ClusterFailOverPolicy` attribute to `Auto`.

The default port for the steward is 14156.

Secure communication in global clusters

In global clusters, VCS provides the option of making the following types of communication secure:

- Communication between the wide-area connectors.
- Communication between the wide-area connectors and the Steward process.

For secure authentication, the wide-area connector process gets a security context as an account in the local authentication broker on each cluster node.

The WAC account belongs to the same domain as HAD and Command Server and is specified as:

```
name = WAC  
domain = VCS_SERVICES@cluster_uuid
```

See [“Cluster attributes”](#) on page 583.

You must configure the wide-area connector process in all clusters to run in secure mode. If the wide-area connector process runs in secure mode, you must run the Steward in secure mode.

See [“Configuring the Steward process \(optional\)”](#) on page 403.

See [“Prerequisites for clusters running in secure mode”](#) on page 396.

Prerequisites for global clusters

This topic describes the prerequisites for configuring global clusters.

Prerequisites for cluster setup

You must have at least two clusters to set up a global cluster. Every cluster must have the required licenses. A cluster can be part of only one global cluster. VCS supports a maximum of four clusters participating in a global cluster.

Clusters must be running on the same platform. The operating system versions must also be the same. Clusters must be using the same VCS version.

Cluster names must be unique within each global cluster; system and resource names need not be unique across clusters. Service group names need not be unique across clusters; however, global service groups must have identical names.

Every cluster must have a valid virtual IP address, which is tied to the cluster. Define this IP address in the cluster’s ClusterAddress attribute. This address is normally configured as part of the initial VCS installation.

All clusters in a global cluster must use either IPv4 or IPv6 addresses. VCS does not support configuring clusters that use different Internet Protocol versions in a global cluster.

For remote cluster operations, you must configure a VCS user with the same name and privileges in each cluster.

See [“User privileges for cross-cluster operations”](#) on page 69.

Prerequisites for application setup

Applications to be configured as global groups must be configured to represent each other in their respective clusters. All application groups in a global group must have the same name in each cluster. The individual resources of the groups can

be different. For example, one group might have a MultiNIC resource or more Mount-type resources. Client systems redirected to the remote cluster in case of a wide-area failover must be presented with the same application they saw in the primary cluster.

However, the resources that make up a global group must represent the same application from the point of the client as its peer global group in the other cluster. Clients redirected to a remote cluster should not be aware that a cross-cluster failover occurred, except for some downtime while the administrator initiates or confirms the failover.

Prerequisites for wide-area heartbeats

There must be at least one wide-area heartbeat going from each cluster to every other cluster. VCS starts communicating with a cluster only after the heartbeat reports that the cluster is alive. VCS uses the ICMP ping by default, the infrastructure for which is bundled with the product. VCS configures the `lcmp` heartbeat if you use Cluster Manager (Java Console) to set up your global cluster. Other heartbeats must be configured manually.

Although multiple heartbeats can be configured but one heartbeat is sufficient to monitor the health of the remote site. Because `lcmp` & `lcmpS` heartbeats use IP network to check the health of the remote site. Even one heartbeat is not a single point of failure if the network is sufficiently redundant. Adding multiple heartbeats will not be useful if they have a single point of failure.

If you have a separate connection for the replication of data between the two sites, then that can be used to reduce single point of failure. Currently, Veritas only ships heartbeat agent for symmetric arrays.

Prerequisites for ClusterService group

The ClusterService group must be configured with the Process (for the wide-area connector), NIC, and IP resources. The service group may contain additional resources for Cluster Management Console and notification, if these components are configured. It is configured automatically when VCS is installed or upgraded.

Prerequisites for replication setup

VCS global clusters are used for disaster recovery, so you must set up real-time data replication between clusters. You can use VCS agents for supported replication solutions to manage the replication.

Prerequisites for clusters running in secure mode

If you plan to configure secure communication among clusters in the global clusters, then you must meet the following prerequisites:

- You must configure the wide area connector processes in both clusters to run in secure mode.
When you configure security using the installer, the installer creates an AT account for the wide-area connector also.
- Both clusters must run in secure mode.
 - You can configure security by using the `installvcs -security` command.
- Both the clusters must share a trust relationship. You can set up a trust relationship by using the `installvcs -securitytrust` command.

Setting up a global cluster

This topic describes how to plan, configure, and test a global cluster. It provides an example of converting a single instance Oracle database configured for local high availability in a VCS cluster to a highly available, disaster-protected infrastructure using a second cluster. The solution uses Volume Replicator to replicate data.

The following figure shows an example of a single-instance Oracle database that is configured as a VCS service group (appgroup) on a two-node cluster.

Figure 13-4 Example: A single-instance Oracle database is configured as a VCS service group (appgroup) on a two-node cluster



Note: Before beginning the process, review the prerequisites and make sure your configuration is ready for a global cluster application: See “[Prerequisites for global clusters](#)” on page 394.

Setting up a global cluster involves the following steps:

- Prepare the application for the global environment
See “[Preparing the application for the global environment](#)” on page 397.
- Configure the ClusterService group
See “[Configuring the ClusterService group](#)” on page 398.
- Configure replication resources in VCS
See “[Configuring replication resources in VCS](#)” on page 399.
- Link the application and replication service groups
See “[Linking the application and replication service groups](#)” on page 401.
- Configure the second cluster
See “[Configuring the second cluster](#)” on page 401.
- Linkclusters
See “[Linking clusters](#)” on page 402.
- Configure the Steward process
See “[Configuring the Steward process \(optional\)](#)” on page 403.
- Configure the global service group
See “[Configuring the global service group](#)” on page 405.

Preparing the application for the global environment

This topic describes how to set up a global cluster environment.

To prepare the application for the global cluster environment

- 1 Install the application (Oracle in this example) in the second cluster.
Make sure the installation is identical with the one in the first cluster.
- 2 Set up replication between the shared disk groups in both clusters.

If your configuration uses Volume Replicator, the process involves grouping the shared data volumes in the first cluster into a Replicated Volume Group (RVG), and creating the Volume Replicator Secondary on hosts in the new cluster, located in your remote site.

See Volume Replicator documentation.

Configuring the ClusterService group

You can configure the service group using the VCS Configuration wizard, Cluster Manager (Java Console), or the command line.

For instructions on how to create the service group using the wizard:

See [“Configuring the ClusterService group”](#) on page 284.

To configure the ClusterService group

- 1 If the ClusterService group does not exist in the cluster create a new service group called ClusterService.
- 2 Add resources of type IP, NIC, and Process to the service group.
- 3 Name the NIC resource csgnic and configure the MACAddress attribute for the resource. The MACAddress is the physical address of the adapter on the system. This attribute has a per-system value.
- 4 Name the IP resource webip and configure the following attributes for the resource:
 - MACAddress—The physical address of the adapter on the system. This attribute could have a per-system value.
 - Address—The virtual IP address for communicating between clusters. The IP address must have a DNS entry.
 - SubNetMask—The subnet mask associated with the virtual IP address.
- 5 Name the Process resource wac and configure the following attributes for the resource:
 - StartProgram—Complete path to the wide-area connector process.
 - If the clusters are running in secure mode, you can set this attribute to: %VCS_HOME%\bin\wac.exe -secure. For example: C:\Program Files\VERITAS\Cluster Server\bin\wac.exe -secure.
 - If the clusters are not running in secure mode, set this attribute to: %VCS_HOME%\bin\wac.exe
For example: C:\Program Files\VERITAS\Cluster Server\bin\wac.exe.
 - StopProgram—Complete path to the program that stops the wac process. Set this attribute to: %VCS_HOME%\bin\wacstop.exe For example: C:\Program Files\VERITAS\Cluster Server\bin\wacstop.exe.
 - MonitorProgram—Complete path to the program that monitors the wac process, typically C:\Program Files\VERITAS\Cluster Server\bin\wacmonitor.exe.

- 6 Mark the `wac` resource as critical.
- 7 Set resource dependencies as per the following information:
 - Process resource (`wac`) depends on the IP resource (`webip`)
 - IP resource (`webip`) depends on the NIC resource (`csgnic`)Enable the resources and bring the `ClusterService` group online.

Configuring replication resources in VCS

This topic describes how to set up replication using Volume Replicator.

VCS supports several replication solutions for global clustering. Contact your Veritas sales representative for the solutions that VCS supports.

About the prerequisites for configuring replication resources in VCS

- Create Replicator Log Volumes for the primary and secondary sites.
- Create the replicated data sets for Volume Replicator. See the Volume Replicator documentation for instructions.
- Verify that the disk group is imported on the node on which you want to create the Volume Replicator RVG Service Group.
- Verify VCS is running, by running the following command on the host on which the you intend to run the Volume Replicator configuration Wizard.

To create a Volume Replicator service group

- 1 From the active node of the cluster at the primary site, click **Start > All Programs > Veritas > Veritas Cluster Server > Configuration Tools > Volume Replicator Agent Configuration Wizard** to launch the configuration wizard.

On Windows 2012 operating systems, use the **Apps** menu.

- 2 Read and verify the requirements on the Welcome panel, and click **Next**.
- 3 In the Wizard Options panel, click **Create a new replication service group**, and then click **Next**.
- 4 Specify the service group name and system priority list:
 - Enter the service group name.
 - In the Available Cluster Systems box, click the nodes on which to configure the service group, and click the right-arrow icon to move the nodes to the service group's system list. Make sure that the set of nodes selected for

the replication service group is the same or a superset of nodes selected for the application's Server service group. Ensure that the nodes are in the same priority order.

- To remove a node from the service group's system list, click the node in the Systems in Priority Orderbox, and click the left arrow icon.
 - To change the priority of a node in the system list, click the node in the Systems in Priority Order box, then click the up and down arrow icons. The node at the top of the list has the highest priority.
 - Click **Next**.
- 5** A message appears, indicating that the configuration will be changed from Read Only to Read/Write. Click **Yes** to continue.
- 6** In the Disk Group and Replicated Volume Group Configuration panel:
- Select **Configure RVGPrimary resource for selected RVG**.
 - Select the replicated volume group for which you want to configure the Volume Replicator RVG resource.
 - Click **Next**.
- 7** In the IP Resource Options panel, select **Create a new IP resource** and click **Next**.
- 8** Enter the network information:
- Verify or enter the virtual IP address; use the IP address specified as the primary IP address when you configured the RDS.
 - Verify the subnet mask.
 - Specify the adapters for each system in the configuration.
 - Click **Next**.

Note: At this step, the specified IP address does not yet need to exist.

- 9** If a message appears, indicating that the specified IP is not configured for replication in this RVG, click **OK** to continue.
- 10** Review the summary of the service group configuration:
- The Resourcesbox lists the configured resources. Click a resource to view its attributes and their configured values in the Attributesbox.
- If necessary, change the resource names; the wizard assigns unique names to resources based on their respective name rules.

To edit a resource name, click the resource name and modify it in the right pane. Press Enter after editing each attribute. To cancel editing a resource name, press Esc.

- Click **Next** to create the Volume Replicator service group.
- 11 When prompted, click **Yes** to create the service group.
Click Finish to bring the replication service group online.

Linking the application and replication service groups

Set an online local hard group dependency from appgroup to appgroup_rep to ensure that the service groups fail over and switch together.

To link the service groups

- 1 In the Cluster Explorer configuration tree, click the cluster name.
- 2 In the view panel, click the **Service Groups** tab. This opens the service group dependency graph.
- 3 Click **Link**.
- 4 Click the parent group, appgroup, and move the mouse toward the child group, appgroup_rep.
- 5 Click the child group appgroup_rep.
- 6 In the Link Service Groups dialog box, click the online local relationship and the hard dependency type and click **OK**.

Configuring the second cluster

This topic describes how to configure a second cluster:

To configure a second cluster

- 1 Modify the ClusterService group in the second cluster for global cluster configuration.

See [“Configuring the ClusterService group”](#) on page 398.

- 2 Create a configuration that is similar to the one in the first cluster.

You can do this by either using Cluster Manager (Java Console) to copy and paste resources from the primary cluster, or by copying the configuration of the appgroup and appgroup_rep groups from the main.cf file in the primary cluster to the secondary cluster.

Run the Volume Replicator Configuration wizard to set up the Volume Replicator service group.

- 3 To assign remote administration privileges to users, configure users with the same name and privileges on both clusters.

See [“User privileges for cross-cluster operations”](#) on page 69.

- 4 Make appropriate changes to the configuration. For example, you must modify the SystemList attribute to reflect the systems in the secondary cluster.

Make sure that the name of the service group (appgroup) is identical in both clusters.

Volume Replicator best practice is to use the same disk group and RVG name on both sites.

If the volume names are the same on both sides, the Mount resources will mount the same block devices, and the same Oracle instance will start at the secondary in case of a failover.

Linking clusters

After the VCS and Volume Replicator infrastructure has been set up at both sites, you must link the two clusters. The Remote Cluster Configuration wizard provides an easy interface to link clusters.

To link clusters

- 1 Verify that the virtual IP address for the ClusterAddress attribute for each cluster is set.

Use the same IP address as the one assigned to the IP resource in the ClusterService group.

- 2 If you are adding a cluster to an existing global cluster environment, run the wizard from a cluster in the global cluster environment. Otherwise, run the wizard from any cluster. From Cluster Explorer, click Edit>Add/Delete Remote Cluster.

See [“Adding a remote cluster”](#) on page 421.

To configure an additional heartbeat between the clusters (optional)

- 1 On Cluster Explorer’s **Edit** menu, click **Configure Heartbeats**.
- 2 In the Heartbeat configuration dialog box, enter the name of the heartbeat and select the check box next to the name of the cluster.
- 3 Click the icon in the **Configure** column to open the Heartbeat Settings dialog box.

- 4 Specify the value of the Arguments attribute and various timeout and interval fields. Click + to add an argument value; click - to delete it.

If you specify IP addresses in the Arguments attribute, make sure the IP addresses have DNS entries.

- 5 Click **OK**.
- 6 Click **OK** in the Heartbeat configuration dialog box.

Now, you can monitor the state of both clusters from the Java Console:

Configuring the Steward process (optional)

In case of a two-cluster GCO, you can configure a Steward to prevent potential split-brain conditions, provided the proper network infrastructure exists.

See [“About the Steward process: Split-brain in two-cluster global clusters”](#) on page 392.

To configure the Steward process for clusters not running in secure mode

- 1 Identify a system that will host the Steward process.

Make sure both clusters can connect to the system through a ping command.

- 2 Copy the file `steward` from a node in the cluster to the Steward system. The file resides at the following path:

```
%VCS_HOME%\bin
```

The variable `%VCS_HOME%` represents the VCS installation directory, typically `C:\Program Files\VERITAS\Cluster Server`.

- 3 In both clusters, set the Stewards attribute to the IP address of the system running the Steward process. For example:

```
cluster cluster1938 (
  UserNames = { admin = gNOgNInKOjOOmWoiNL }
  ClusterAddress = "10.182.147.19"
  Administrators = { admin }
  CredRenewFrequency = 0
  CounterInterval = 5
  Stewards = {"10.212.100.165"}
}
```

- 4 On the system designated to host the Steward, start the Steward process:

```
steward.exe -start
```

To configure the Steward process for clusters running in secure mode

- 1 Verify the prerequisites for securing Steward communication are met.
See “[Prerequisites for clusters running in secure mode](#)” on page 396.

- 2 Identify a system that will host the Steward process.
Make sure both clusters can connect to the system through a ping command.

- 3 Copy the steward file from a node in the cluster to the Steward system. The file resides at the following path:

```
%VCS_HOME%\bin\
```

The variable %VCS_HOME% represents the VCS installation directory, typically

.

- 4 On the Steward system, run the following script to generate certificates for the steward:

```
"C:\Program Files\Veritas\VRTSPerl\bin\perl.exe" create_certs.pl
```

This script is located in the %VCS_HOME%\bin directory.

- 5 To establish trust, run the following script on these systems:

- On the Steward system:

```
"C:\Program Files\Veritas\VRTSPerl\bin\perl.exe" setup_trust.pl  
virtualIP STEWARD
```

Here, *virtualIP* is the virtual IP address of the cluster.

- On any node in a global cluster:

```
"C:\Program Files\Veritas\VRTSPerl\bin\perl.exe" setup_trust.pl
```

If it is multi-site cluster, run this script on any cluster node at each site.

- On each node in the cluster:

```
"C:\Program Files\Veritas\VRTSPerl\bin\perl.exe" setup_trust.pl  
stewardNameOrIP WAC
```

Here, *stewardNameOrIP* is the system name or IP address of the Steward system.

This script is located in the %VCS_HOME%\bin directory.

- 6 On the Steward system, create a file called `Steward.conf` and populate it with the following information:

```
broker=system_name
accountname=accountname
domain=HA_SERVICES@FQDN_of_system_that_issued_the_certificate
```

- 7 In both clusters, set the Stewards attribute to the IP address of the system that runs the Steward process. For example:

```
cluster cluster1938 {
  UserNames = { admin = gNOgNInKOjOOmWOiNL }
  ClusterAddress = "10.182.147.19"
  Administrators = { admin }
  CredRenewFrequency = 0
  CounterInterval = 5
  Stewards = {"10.212.100.165"}
}
```

- 8 On the system designated to run the Steward, start the Steward process:

```
steward.exe -start -secure
```

Stopping the Steward process

When you start the Steward, the process does not release the command window. Stop the Steward process, by typing control+C in the command window or open another command window and run the command to stop the Steward process.

To stop the Steward process that is not configured in secure mode

- ◆ Open a new command window and run the following command:

```
steward.exe -stop
```

To stop the Steward process running in secure mode

- ◆ Open a new command window and run the following command:

```
steward.exe -stop -secure
```

Configuring the global service group

Configure the Oracle service group, appgroup, as a global group by running the Global Group Configuration wizard.

To create the global service group

- 1** In the service group tree of Cluster Explorer, right-click the application service group (appgroup)
- 2** Select **Configure As Global** from the menu.
- 3** Enter the details of the service group to modify (appgroup).
- 4** From the **Available Clusters** box, click the clusters on which the group can come online. The local cluster is not listed as it is implicitly defined to be part of the ClusterList. Click the right arrow to move the cluster name to the **ClusterList** box.
- 5** Select the policy for cluster failover:
 - **Manual** prevents a group from automatically failing over to another cluster.
 - **Auto** enables a group to automatically fail over to another cluster if it is unable to fail over within the cluster, or if the entire cluster faults.
 - **Connected** enables a group to automatically fail over to another cluster if it is unable to fail over within the cluster.
- 6** Click **Next**.
- 7** Enter or review the connection details for each cluster. Click the **Configure** icon to review the remote cluster information for each cluster.
- 8** Enter the IP address of the remote cluster, the IP address of a cluster system, or the host name of a cluster system.
- 9** Enter the user name and the password for the remote cluster and click **OK**.
- 10** Click **Next**.
- 11** Click **Finish**.
- 12** Save the configuration.

The appgroup service group is now a global group and can be failed over between clusters.

For remote cluster operations, you must configure a VCS user with the same name and privileges in each cluster.

See [“User privileges for cross-cluster operations”](#) on page 69.

About cluster faults

In the global cluster setup, consider a case where the primary cluster suffers a failure. The Oracle service group cannot fail over in the local cluster and must fail over globally, to a node in another cluster.

In this situation, VCS sends an alert indicating that the cluster is down.

An administrator can bring the group online in the remote cluster.

The RVGPrimary agent ensures that Volume Replicator volumes are made writable and the DNS agent ensures that name services are resolved to the remote site. The application can be started at the remote site.

About the type of failure

If a disaster disables all processing power in your primary data center, heartbeats from the failover site to the primary data center fail. VCS sends an alert signalling cluster failure. If you choose to take action on this failure, VCS prompts you to declare the type of failure.

You can choose one of the following options to declare the failure:

- Disaster, implying permanent loss of the primary data center
- Outage, implying the primary may return to its current form in some time
- Disconnect, implying a split-brain condition; both clusters are up, but the link between them is broken
- Replica, implying that data on the takeover target has been made consistent from a backup source and that the RVGPrimary can initiate a takeover when the service group is brought online. This option applies to Volume Replicator environments only.

You can select the groups to be failed over to the local cluster, in which case VCS brings the selected groups online on a node based on the group's FailOverPolicy attribute. It also marks the groups as being OFFLINE in the other cluster. If you do not select any service groups to fail over, VCS takes no action except implicitly marking the service groups as offline in the failed cluster.

Switching the service group back to the primary

You can switch the service group back to the primary after resolving the fault at the primary site. Before switching the application to the primary site, you must resynchronize any changed data from the active Secondary site since the failover. This can be done manually through Volume Replicator or by running a VCS action from the RVGPrimary resource.

To switch the service group when the primary site has failed and the secondary did a takeover

- 1 In the **Service Groups** tab of the configuration tree, right-click the resource.
- 2 Click **Actions**.
- 3 Specify the details of the action:

- From the **Action** list, choose fbsync.
- Click the system on which to execute the action.
- Click **OK**.

This begins a fast-failback of the replicated data set. You can monitor the value of the ResourceInfo attribute for the RVG resource to determine when the resynchronization has completed.

- 4 Once the resynchronization completes, switch the service group to the primary cluster.
 - In the **Service Groups** tab of the Cluster Explorer configuration tree, right-click the service group.
 - Click **Switch To**, and click **Remote switch**.
 - In the Switch global group dialog box, click the cluster to switch the group. Click the specific system, or click **Any System**, and click **OK**.

About setting up a disaster recovery fire drill

The disaster recovery fire drill procedure tests the fault-readiness of a configuration by mimicking a failover from the primary site to the secondary site. This procedure is done without stopping the application at the primary site and disrupting user access, interrupting the flow of replicated data, or causing the secondary site to need resynchronization.

The initial steps to create a fire drill service group on the secondary site that closely follows the configuration of the original application service group and contains a point-in-time copy of the production data in the Replicated Volume Group (RVG). Bringing the fire drill service group online on the secondary site demonstrates the ability of the application service group to fail over and come online at the secondary site, should the need arise. Fire drill service groups do not interact with outside clients or with other instances of resources, so they can safely come online even when the application service group is online.

You must conduct a fire drill only at the secondary site; do not bring the fire drill service group online on the node hosting the original application.

You can override the FireDrill attribute and make fire drill resource-specific.

Before you perform a fire drill in a disaster recovery setup that uses Volume Replicator, perform the following steps:

- Configure the fire drill service group.
See [“About creating and configuring the fire drill service group manually”](#) on page 409.
- Set the value of the ReuseMntPt attribute to 1 for all Mount resources.
- After the fire drill service group is taken offline, reset the value of the ReuseMntPt attribute to 0 for all Mount resources.

Set an offline local dependency between the fire drill service group and the application service group to make sure a fire drill does not block an application failover in case a disaster strikes the primary site.

See [“About creating and configuring the fire drill service group manually”](#) on page 409.

VCS also supports HA fire drills to verify a resource can fail over to another node in the cluster.

For detailed instructions on how to set up a fire drill in using the Solutions Configurations Center, see the following documents:

- *Storage Foundation and High Availability Solutions HA and Disaster Recovery Solutions Guide for Microsoft SQL*
- *Storage Foundation and High Availability Solutions HA and Disaster Recovery Solutions Guide for Microsoft Exchange*
- *Storage Foundation and High Availability Solutions, Solutions Guide*

About creating and configuring the fire drill service group manually

You can create the fire drill service group using the command line or Cluster Manager (Java Console.) The fire drill service group uses the duplicated copy of the application data.

Creating and configuring the fire drill service group involves the following tasks:

- See [“Creating the fire drill service group”](#) on page 410.
- See [“Linking the fire drill and replication service groups”](#) on page 410.
- See [“Adding resources to the fire drill service group”](#) on page 411.
- See [“Configuring the fire drill service group”](#) on page 411.
- See [“Enabling the FireDrill attribute”](#) on page 412.

Creating the fire drill service group

This section describes how to use the Cluster Manager (Java Console) to create the fire drill service group and change the failover attribute to false so that the fire drill service group does not failover to another node during a test.

To create the fire drill service group

- 1 Open the Veritas Cluster Manager (Java Console). (**Start > All Programs > Veritas > Veritas Cluster Manager - Java Console**)
On Windows 2012 operating systems, use the **Apps** menu.
- 2 Log on to the cluster and click **OK**.
- 3 Click the **Service Group** tab in the left pane and click the **Resources** tab in the right pane.
- 4 Right-click the cluster in the left pane and click **Add Service Group**.
- 5 In the **Add Service Group** dialog box, provide information about the new service group.
 - In Service Group name, enter a name for the fire drill service group
 - Select systems from the Available Systems box and click the arrows to add them to the Systems for Service Group box.
 - Click **OK**.

To disable the AutoFailOver attribute

- 1 Click the **Service Group** tab in the left pane and select the fire drill service group.
- 2 Click the **Properties** tab in the right pane.
- 3 Click the **Show all attributes** button.
- 4 Double-click the **AutoFailOver** attribute.
- 5 In the **Edit Attribute** dialog box, clear the **AutoFailOver** check box.
- 6 Click **OK** to close the **Edit Attribute** dialog box.
- 7 Click the **Save and Close Configuration** icon in the tool bar.

Linking the fire drill and replication service groups

Create an online local firm dependency link between the fire drill service group and the replication service group.

To link the service groups

- 1 In Cluster Explorer, click the System tab in the left pane and click the **Service Groups** tab in the right pane.
- 2 Click **Link**.
- 3 Click the fire drill service group, drag the link and click the replication service group.
- 4 Define the dependency. Choose the **online local** and **firm** options and click **OK**.

Adding resources to the fire drill service group

Add resources to the new fire drill service group to recreate key aspects of the application service group.

To add resources to the service group

- 1 In Cluster Explorer, click the **Service Group** tab in the left pane, click the application service group and click the **Resources** tab in the right pane.
- 2 Right-click the resource at the top of the tree, select **Copy** and click **Self and Child Nodes**.
- 3 In the left pane, click the fire drill service group.
- 4 Right-click the right pane, and click **Paste**.
- 5 In the Name Clashes dialog box, specify a way for the resource names to be modified, for example, insert an FD_ prefix. Click **Apply**.
- 6 Click **OK**.

Configuring the fire drill service group

After copying resources to the fire drill service group, edit the resources so they will work properly with the duplicated data. The attributes must be modified to reflect the configuration at the remote site. Bringing the service group online without modifying resource attributes is likely to result in a cluster fault and interruption in service.

To configure the service group

- 1 In Cluster Explorer, click the **Service Group** tab in the left pane, click the fire drill service group in the left pane and click the **Resources** tab in the right pane.
- 2 Right-click the RVGPrimary resource and click **Delete**.

- 3 Right-click the resource to be edited and click **View > Properties View**. If a resource to be edited does not appear in the pane, click **Show All Attributes**.
- 4 Edit attributes to reflect the configuration at the remote site. For example, change the MountV resources so that they point to the volumes used in the fire drill service group. Similarly, reconfigure the DNS and IP resources.

Enabling the FireDrill attribute

You must edit certain resource types so they are FireDrill-enabled. Making a resource type FireDrill-enabled changes the way that VCS checks for concurrency violations. Typically, when FireDrill is not enabled, resources can not come online on more than one node in a cluster at a time. This behavior prevents multiple nodes from using a single resource or from answering client requests. Fire drill service groups do not interact with outside clients or with other instances of resources, so they can safely come online even when the application service group is online.

Typically, you would enable the FireDrill attribute for the resource type that are used to configure the agent.

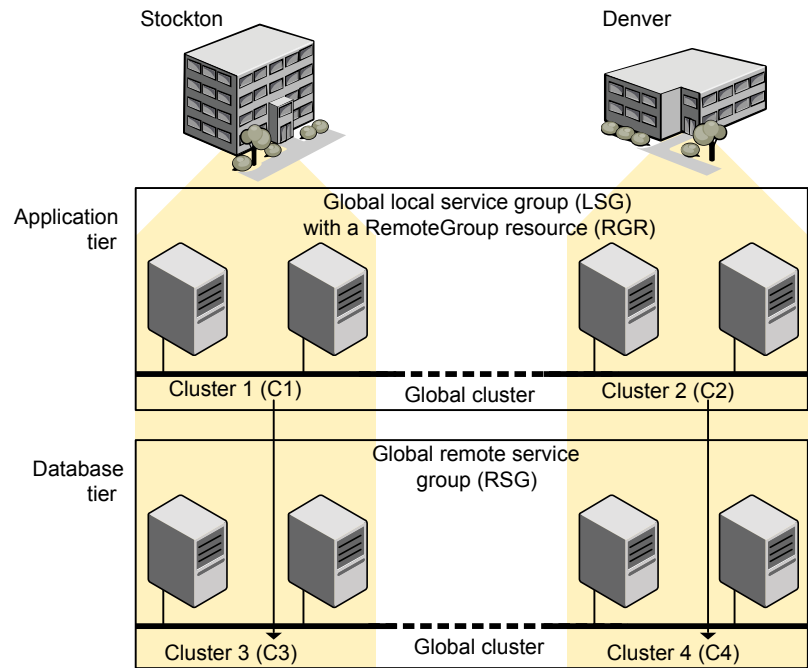
To enable the FireDrill attribute

- 1 In Cluster Explorer, click the **Types** tab in the left pane, right-click the type to be edited, and click **View > Properties View**.
- 2 Click **Show All Attributes**.
- 3 Double click **FireDrill**.
- 4 In the **Edit Attribute** dialog box, enable **FireDrill** as required, and click **OK**.
Repeat the process of enabling the FireDrill attribute for all required resource types.

Multi-tiered application support using the RemoteGroup agent in a global environment

Figure 13-5 represents a two-site, two-tier environment. The application cluster, which is globally clustered between L.A. and Denver, has cluster dependencies up and down the tiers. Cluster 1 (C1), depends on the remote service group for cluster 3 (C3). At the same time, cluster 2 (C2) also depends on the remote service group for cluster 4 (C4).

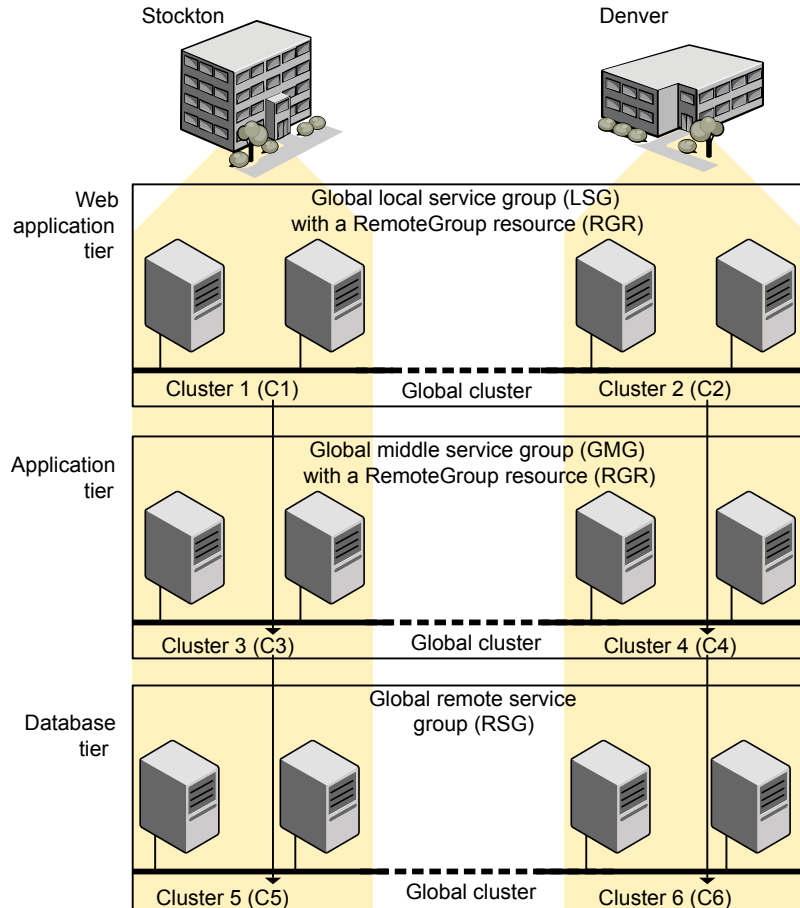
Figure 13-5 A VCS two-tier globally clustered application and database



Just as a two-tier, two-site environment is possible, you can also tie a three-tier environment together.

Figure 13-6 represents a two-site, three-tier environment. The application cluster, which is globally clustered between L.A. and Denver, has cluster dependencies up and down the tiers. Cluster 1 (C1), depends on the RemoteGroup resource on the DB tier for cluster 3 (C3), and then on the remote service group for cluster 5 (C5). The stack for C2, C4, and C6 functions the same.

Figure 13-6 A three-tiered globally clustered application, database, and storage



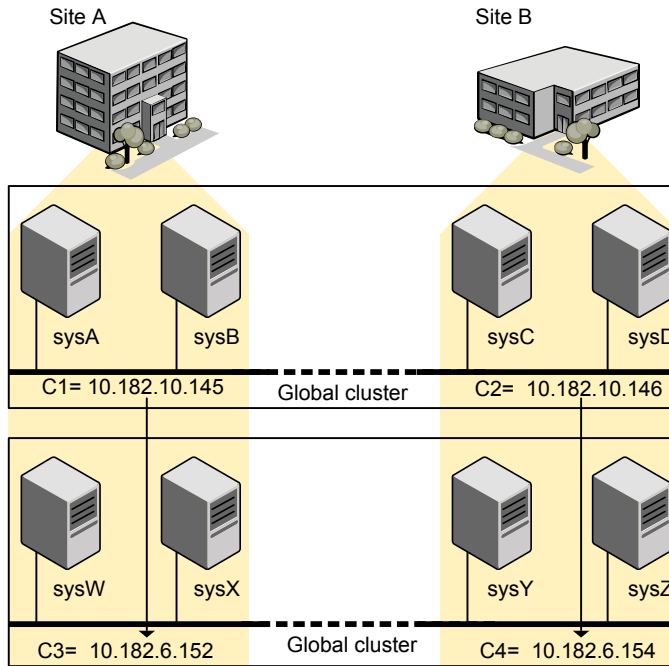
Test scenario for a multi-tiered environment

In the following scenario, eight systems reside in four clusters. Each tier contains a global cluster. The global local service group in the top tier depends on the global remote service group in the bottom tier.

The following main.cf files show this multi-tiered environment. The FileOnOff resource is used to test the dependencies between layers. Note that some attributes have been edited for clarity, and that these clusters are not running in secure mode.

Figure 13-7 shows the scenario for testing.

Figure 13-7 A VCS two-tiered globally clustered scenario



About the main.cf file for cluster 1

The contents of the main.cf file for cluster 1 (C1) in the top tier, containing the sysA and sysB nodes.

```
include "types.cf"

cluster C1 (
    ClusterAddress = "10.182.10.145"
)

remotecluster C2 (
    ClusterAddress = "10.182.10.146"
)

heartbeat Icmp (
    ClusterList = { C2 }
    AYATimeout = 30
    Arguments @C2 = { "10.182.10.146" }
```

```

    )

system sysA (
    )

system sysB (
    )

group LSG (
    SystemList = { sysA = 0, sysB = 1 }
    ClusterList = { C2 = 0, C1 = 1 }
    AutoStartList = { sysA, sysB }
    ClusterFailOverPolicy = Auto
    )

FileOnOff filec1 (
    PathName = "/tmp/c1"
    )

RemoteGroup RGR (
    IPAddress = "10.182.6.152"
    // The above IPAddress is the highly available address of C3—
    // the same address that the wac uses
    Username = root
    Password = xxxyyy
    GroupName = RSG
    VCSSysName = ANY
    ControlMode = OnOff
    )

```

About the main.cf file for cluster 2

The contents of the main.cf file for cluster 2 (C2) in the top tier, containing the sysC and sysD nodes.

```

include "types.cf"

cluster C2 (
    ClusterAddress = "10.182.10.146"
    )

remotecluster C1 (
    ClusterAddress = "10.182.10.145"

```



```

    )

    heartbeat Icmp (
        ClusterList = { C1 }
        AYATimeout = 30
        Arguments @C1 = { "10.182.10.145" }
    )

    system sysC (
    )

    system sysD (
    )

    group LSG (
        SystemList = { sysC = 0, sysD = 1 }
        ClusterList = { C2 = 0, C1 = 1 }
        Authority = 1
        AutoStartList = { sysC, sysD }
        ClusterFailOverPolicy = Auto
    )

    FileOnOff filec2 (
        PathName = filec2
    )

    RemoteGroup RGR (
        IPAddress = "10.182.6.154"
        // The above IPAddress is the highly available address of C4—
        // the same address that the wac uses
        Username = root
        Password = vvvyyy
        GroupName = RSG
        VCSSysName = ANY
        ControlMode = OnOff
    )

```

About the main.cf file for cluster 3

The contents of the main.cf file for cluster 3 (C3) in the bottom tier, containing the sysW and sysX nodes.

```
include "types.cf"

cluster C3 (
    ClusterAddress = "10.182.6.152"
)

remotecluster C4 (
    ClusterAddress = "10.182.6.154"
)

heartbeat Icmp (
    ClusterList = { C4 }
    AYATimeout = 30
    Arguments @C4 = { "10.182.6.154" }
)

system sysW (
)

system sysX (
)

group RSG (
    SystemList = { sysW = 0, sysX = 1 }
    ClusterList = { C3 = 1, C4 = 0 }
    AutoStartList = { sysW, sysX }
    ClusterFailOverPolicy = Auto
)

FileOnOff filec3 (
    PathName = "/tmp/filec3"
)
```

About the main.cf file for cluster 4

The contents of the main.cf file for cluster 4 (C4) in the bottom tier, containing the sysY and sysZ nodes.

```
include "types.cf"

cluster C4 (
    ClusterAddress = "10.182.6.154"
)
```

```
remotecluster C3 (  
    ClusterAddress = "10.182.6.152"  
)  
  
heartbeat Icmp (  
    ClusterList = { C3 }  
    AYATimeout = 30  
    Arguments @C3 = { "10.182.6.152" }  
)  
  
system sysY (  
)  
  
system sysZ (  
)  
  
group RSG (  
    SystemList = { sysY = 0, sysZ = 1 }  
    ClusterList = { C3 = 1, C4 = 0 }  
    Authority = 1  
    AutoStartList = { sysY, sysZ }  
    ClusterFailOverPolicy = Auto  
)  
  
FileOnOff filec4 (  
    PathName = "/tmp/filec4"  
)
```

Administering global clusters from Cluster Manager (Java console)

This chapter includes the following topics:

- [About global clusters](#)
- [Adding a remote cluster](#)
- [Deleting a remote cluster](#)
- [Administering global service groups](#)
- [Administering global heartbeats](#)

About global clusters

The process of creating a global cluster environment involves creating a common service group for specified clusters, making sure all the service groups are capable of being brought online in the specified clusters, connecting the standalone clusters, and converting the service group that is common to all the clusters to a global service group. Use the console to add and delete remote clusters, create global service groups, and manage cluster heartbeats.

Creating a global cluster environment requires the following conditions:

- All service groups are properly configured and able to come online.
- The service group that will serve as the global group has the same unique name across all applicable clusters.
- The clusters must use the same version of VCS.

- The clusters must use the same operating system.
- The clusters are standalone and do not already belong to a global cluster environment.

Through the Java Console, you can simulate the process of generating and clearing global cluster faults in an OFFLINE state. Use VCS Simulator to complete these operations.

For remote cluster operations, you must configure a VCS user with the same name and privileges in each cluster.

See [“User privileges for cross-cluster operations”](#) on page 69.

Adding a remote cluster

Cluster Explorer provides a wizard to create global clusters by linking standalone clusters. Command Center only enables you to perform remote cluster operations on the local cluster.

- If you are creating a global cluster environment for the first time with two standalone clusters, run the wizard from either of the clusters.
- If you are adding a standalone cluster to an existing global cluster environment, run the wizard from a cluster already in the global cluster environment.

The following information is required for the Remote Cluster Configuration Wizard in Cluster Explorer:

- The active host name or IP address of each cluster in the global configuration and of the cluster being added to the configuration.
- The user name and password of the administrator for each cluster in the configuration.
- The user name and password of the administrator for the cluster being added to the configuration.

Note: Veritas InfoScale products do not support adding a cluster that is already part of a global cluster environment. To merge the clusters of one global cluster environment (for example, cluster A and cluster B) with the clusters of another global environment (for example, cluster C and cluster D), separate cluster C and cluster D into standalone clusters and add them one by one to the environment containing cluster A and cluster B.

To add a remote cluster to a global cluster environment in Cluster Explorer

- 1** Do one of the following to add a remote cluster to a global cluster environment in Cluster Explorer:

From Cluster Explorer, click **Add/Delete Remote Cluster** on the **Edit** menu.

or

From the Cluster Explorer configuration tree, right-click the cluster name, and click **Add/Delete Remote Clusters**.

- 2** Review the required information for the **Remote Cluster Configuration Wizard** and click **Next**.

- 3** In the Wizard Options dialog box, click **Add Cluster** and then, click **Next**.

- 4** Enter the details of the new cluster:

If the cluster is not running in secure mode, do the following:

- Enter the host name of a cluster system, an IP address of a cluster system, or the IP address of the cluster that will join the global environment.
- Verify the port number.
- Enter the user name and the password.
- Click **Next**.

If the cluster is running in secure mode, do the following:

- Enter the host name of a cluster system, an IP address of a cluster system, or the IP address of the cluster that will join the global environment.
- Verify the port number.
- Choose to connect to the remote cluster with the credentials used for the current cluster connection or enter new credentials, including the user name, password, and the domain.

If you have connected to the remote cluster using the wizard earlier, you can use the credentials from the previous connection.

Click **Next**.

- 5** Enter the details of the existing remote clusters; this information on administrator rights enables the wizard to connect to all the clusters and make changes to the configuration.

- 6** Click the **Configure** icon.

If the cluster is not running in secure mode, do the following:

- Enter the host name of a cluster system, an IP address of a cluster system, or the IP address of the cluster that will join the global environment.

- Verify the port number.
- Enter the user name.
- Enter the password.
- Click **OK**.
- Repeat these steps for each cluster in the global environment.

If the cluster is running in secure mode, do the following:

- Enter the host name of a cluster system, an IP address of a cluster system, or the IP address of the cluster that will join the global environment.
- Verify the port number.
- Choose to connect to the remote cluster with the credentials used for the current cluster connection or enter new credentials, including the user name, password, and the domain.
- Click **OK**.

7 Click **Next**.

8 Click **Finish**. After running the wizard, the configurations on all the relevant clusters are opened and changed; the wizard does not close the configurations.

To add a remote cluster to a global cluster environment in Command Center

- 1 Click **Commands > Configuration > Cluster Objects > Add Remote Cluster**.
- 2 Enter the name of the cluster.
- 3 Enter the IP address of the cluster.
- 4 Click **Apply**.

Note: Command Center enables you to perform operations on the local cluster; this does not affect the overall global cluster configuration.

Deleting a remote cluster

The Remote Cluster Configuration Wizard enables you to delete a remote cluster.

This operation involves the following tasks:

- Taking the ApplicationProcess resource configured to monitor the wac resource offline on the cluster that will be removed from the global environment. For example, to delete cluster C2 from a global environment containing C1 and C2, log on to C2 and take the wac resource offline.

- Removing the name of the specified cluster (C2) from the cluster lists of the other global groups using the Global Group Configuration Wizard. Note that the Remote Cluster Configuration Wizard in Cluster Explorer updates the cluster lists for heartbeats. Log on to the local cluster (C1) to complete this task before using the Global Group Configuration Wizard.
- Deleting the cluster (C2) from the local cluster (C1) using the Remote Cluster Configuration Wizard.

Note: You cannot delete a remote cluster if the cluster is part of a cluster list for global service groups or global heartbeats, or if the cluster is in the **RUNNING**, **BUILD**, **INQUIRY**, **EXITING**, or **TRANSITIONING** states.

To take the **wac** resource offline

- 1 From Cluster Monitor, log on to the cluster that will be deleted from the global cluster environment.
- 2 Do one of the following:

In the **Service Groups** tab of the Cluster Explorer configuration tree, right-click the **wac** resource under the **Process** type in the **ClusterService** group.

or

Click the ClusterService group in the configuration tree, click the **Resources** tab, and right-click the resource in the view panel.
- 3 Click **Offline**, and click the appropriate system from the menu.

To remove a cluster from a cluster list for a global group

- 1 From Cluster Explorer, click **Configure Global Groups** on the **Edit** menu.
- 2 Click **Next**.
- 3 Enter the details of the service group to modify, as follows:
 - Click the name of the service group.
 - For global to local cluster conversion, click the left arrow to move the cluster name from the cluster list back to the **Available Clusters** box.
 - Click **Next**.
- 4 Enter or review the connection details for each cluster. Click the **Configure** icon to review the remote cluster information for each cluster.

If the cluster is not running in secure mode, do the following:
 - Enter the IP address of the remote cluster, the IP address of a cluster system, or the host name of a cluster system.

- Verify the port number.
- Enter the user name.
- Enter the password.
- Click **OK**.

If the cluster is running in secure mode, do the following:

- Enter the IP address of the remote cluster, the IP address of a cluster system, or the host name of a cluster system.
- Verify the port number.
- Choose to connect to the remote cluster using the connected cluster's credentials or enter new credentials, including the user name, password, and the domain.
- Click **OK**.

5 Click **Next**.

6 Click **Finish**.

To delete a remote cluster from the local cluster

1 Do one of the following:

From Cluster Explorer, click **Add/Delete Remote Cluster** on the **Edit** menu.

or

From the Cluster Explorer configuration tree, right-click the cluster name, and click **Add/Delete Remote Clusters**.

2 Review the required information for the **Remote Cluster Configuration Wizard** and click **Next**.

3 In the Wizard Options dialog box, click **Delete Cluster** and click **Next**:

4 In the Delete Cluster dialog box, click the name of the remote cluster to delete, and then click **Next**:

5 Review the connection details for each cluster. Click the **Configure** icon to review the remote cluster information for each cluster.

If the cluster is not running in secure mode, do the following:

- Enter the IP address of the remote cluster, the IP address of a cluster system, or the host name of a cluster system.
- Verify the port number.
- Enter the user name.
- Enter the password.

- Click **OK**.

If the cluster is running in secure mode, do the following:

- Enter the IP address of the remote cluster, the IP address of a cluster system, or the host name of a cluster system.
- Verify the port number.
- Choose to connect to the remote cluster with the credentials used for the current cluster connection or enter new credentials, including the user name, password, and the domain.
- If you have connected to the remote cluster using the wizard earlier, you can use the credentials from the previous connection.
- Click **OK**.

- 6 Click **Finish**.

Administering global service groups

After connecting clusters in a global cluster environment, use the Global Group Configuration Wizard to convert a local service group that is common to the global clusters to a global group. This wizard also enables you to convert global groups into local groups.

Administering global groups requires the following conditions:

- A group that will serve as the global group must have the same name across all applicable clusters.
- You must know the user name and password for the administrator for each cluster in the configuration.

Use Cluster Explorer to bring a global group online and take a global group offline on a remote cluster.

Converting local and global groups

Perform the following procedure to convert local and global groups.

To convert local and global groups

1 Do one of the following:

From Cluster Explorer, click **Configure Global Groups...** on the **Edit** menu.

or

From the Cluster Explorer configuration tree, right-click the service group, click **Configure As Global...** or **Make Local...** and proceed to **3**.

2 Review the information required for the Global Group Configuration Wizard and click **Next**.

3 Enter the details of the service group to modify:

- Click the name of the service group that will be converted from a local group to a global group, or vice versa.
- From the **Available Clusters** box, click the clusters on which the group can come online. Click the right arrow to move the cluster name to the **Clusters for Service Group** box; for global to local cluster conversion, click the left arrow to move the cluster name back to the **Available Clusters** box. A priority number (starting with 0) indicates the cluster in which the group will attempt to come online. If necessary, double-click the entry in the **Priority** column to enter a new value.
- Select one of the following policies for cluster failover:
 - **Manual** prevents a group from automatically failing over to another cluster.
 - **Auto** enables a group to automatically fail over to another cluster if it is unable to fail over within the cluster, or if the entire cluster faults.
 - **Connected** enables a group to automatically fail over to another cluster if it is unable to fail over within the cluster.
- Click **Next**.

4 Enter or review the connection details for each cluster:

Click the **Configure** icon to review the remote cluster information for each cluster.

If the cluster is not running in secure mode, do the following:

- Enter the IP address of the remote cluster, the IP address of a cluster system, or the host name of a cluster system.
- Verify the port number.
- Enter the user name and password.

- Click **OK**.

Repeat these steps for each cluster in the global environment.

If the cluster is running in secure mode, do the following:

- Enter the IP address of the remote cluster, the IP address of a cluster system, or the host name of a cluster system.
- Verify the port number.
- Choose to connect to the remote cluster with the credentials used for the current cluster connection, or enter new credentials, including the user name, password, and the domain.
If you have connected to the remote cluster using the wizard earlier, you can use the credentials from the previous connection.

- Click **OK**.

Repeat these steps for each cluster in the global environment.

- 5 In the Remote cluster information dialog box, click **Next**.
- 6 Click **Finish**.

Bringing a service group online in a remote cluster

This topic describes how to bring a service group online in a remote cluster.

To bring a service group online in a remote cluster

- 1 Do the following:
In the **Service Groups** tab of the Cluster Explorer configuration tree of a local cluster, right-click the service group.
or
Click a local 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 **Remote online...**
- 3 In the Online global group dialog box, do the following:
 - Click the remote cluster to bring the group online.
 - Click the specific system, or click **Any System**, to bring the group online.
 - Click **OK**.
- 4 In the Question dialog box, click **Yes**.

Taking a service group offline in a remote cluster

This topic describes how to take a service group offline in a remote cluster.

To take a service group offline in a remote cluster

- 1 Do the following:

In the **Service Groups** tab of the Cluster Explorer configuration tree of a local cluster, right-click the service group.

or

Click a local 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 **Remote offline...**
- 3 In the Offline global group dialog box, do the following:
 - Click the remote cluster to take the group offline.
 - Click the specific system, or click **All Systems**, to take the group offline.
 - Click **OK**.
- 4 In the Question dialog box, click **Yes**.

Switching a service group to a remote cluster

This topic describes how to switch a service group to a remote cluster.

To switch a service group to a remote cluster

- 1 Do the following:

In the **Service Groups** tab of the Cluster Explorer configuration tree of a local cluster, right-click the service group.

or

Click a local 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 **Remote switch...**
- 3 In the Switch global group dialog box:
 - Click the cluster to switch the group.
 - Click the specific system, or click **Any System**, to switch the group.
- 4 In the Question dialog box, click **Yes**.

Administering global heartbeats

Use Cluster Explorer to add, modify, and delete heartbeats in a global cluster environment. Icmp heartbeats send Icmp packets simultaneously to all IP addresses; IcmpS heartbeats send individual Icmp packets to IP addresses in serial order. Global clustering requires a minimum of one heartbeat between clusters; the Icmp heartbeat is added when the cluster is added to the environment. You can add additional heartbeats as a precautionary measure.

Adding a global heartbeat

This topic describes how to add a global heartbeat.

To add a cluster heartbeat from Cluster Explorer

- 1 Click **Configure Heartbeats** on the **Edit** menu.
- 2 In the Heartbeat Configuration dialog box, do the following:
 - Enter the name of the heartbeat.
 - Select the check box next to the name of the cluster to add it to the cluster list for the heartbeat.
 - Click the icon in the **Configure** column to open the Heartbeat Settings dialog box.
 - Specify the value of the Arguments attribute and various timeout and interval fields. Click **+** to add an argument value; click **-** to delete it.
 - Click **OK**.
 - Click **OK** on the Heartbeat configuration dialog box.

To add a cluster heartbeat from Command Center

- 1 Click **Commands>Configuration>Cluster Objects>Add Heartbeat**.
- 2 Enter the name of the heartbeat.
- 3 Click **Apply**.

Modifying a global heartbeat

This topic describes how to modify a global heartbeat.

To modify a global heartbeat

- 1 From Cluster Explorer, click **Configure Heartbeats** on the **Edit** menu.
- 2 In the Heartbeat Configuration dialog box:
 - Click **Existing Heartbeat**.

- Click the name of the existing heartbeat from the menu.
- Select or clear the check box next to the name of a cluster to add or remove it from the cluster list for the heartbeat.
- If necessary, click the icon in the **Configure** column to open the Heartbeat Settings dialog box. Otherwise, proceed to the last step.
- Change the values of the Arguments attribute and various timeout and interval fields. Click **+** to add an argument value; click **-** to delete it.
- Click **OK**.
- Click **OK** on the Heartbeat Configuration dialog box.

Deleting a global heartbeat

This topic describes how to delete a global heartbeat. You cannot delete the last heartbeat between global clusters.

To delete a cluster heartbeat from Command Center

- 1** Click **Commands>Configuration>Cluster Objects>Delete Heartbeat**.
- 2** Click the heartbeat to delete.
- 3** Click **Apply**.

Administering global clusters from the command line

This chapter includes the following topics:

- [About administering global clusters from the command line](#)
- [About global querying in a global cluster setup](#)
- [Administering global service groups in a global cluster setup](#)
- [Administering resources in a global cluster setup](#)
- [Administering clusters in global cluster setup](#)
- [Administering heartbeats in a global cluster setup](#)

About administering global clusters from the command line

For remote cluster operations, you must configure a VCS user with the same name and privileges in each cluster.

See [“User privileges for cross-cluster operations”](#) on page 69.

Review the following procedures to administer global clusters from the command-line.

See [“About global querying in a global cluster setup”](#) on page 433.

See [“Administering global service groups in a global cluster setup”](#) on page 440.

See [“Administering resources in a global cluster setup”](#) on page 442.

See [“Administering clusters in global cluster setup”](#) on page 442.

See [“Administering heartbeats in a global cluster setup”](#) on page 445.

About global querying in a global cluster setup

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

See [“Querying global cluster service groups”](#) on page 433.

See [“Querying resources across clusters”](#) on page 434.

See [“Querying systems”](#) on page 436.

See [“Querying clusters”](#) on page 436.

See [“Querying status”](#) on page 438.

See [“Querying heartbeats”](#) on page 438.

Querying global cluster service groups

This topic describes how to perform a query on global cluster service groups:

To display service group attribute values across clusters

- ◆ Use the following command to display service group attribute values across clusters:

```
hagrp -value service_group attribute [system]  
[-clus cluster | -localclus]
```

The option `-clus` displays the attribute value on the cluster designated by the variable *cluster*; the option `-localclus` specifies the local cluster.

If the attribute has local scope, you must specify the system name, except when querying the attribute on the system from which you run the command.

To display the state of a service group across clusters

- ◆ Use the following command to display the state of a service group across clusters:

```
hagrp -state [service_groups -sys systems]  
[-clus cluster | -localclus]
```

The option `-clus` displays the state of all service groups on a cluster designated by the variable *cluster*; the option `-localclus` specifies the local cluster.

To display service group information across clusters

- ◆ Use the following command to display service group information across clusters:

```
hagrp -display [service_groups] [-attribute attributes]  
[-sys systems] [-clus cluster | -localclus]
```

The option `-clus` applies to global groups only. If the group is local, the cluster name must be the local cluster name, otherwise no information is displayed.

To display service groups in a cluster

- ◆ Use the following command to display service groups in a cluster:

```
hagrp -list [conditionals] [-clus cluster | -localclus]
```

The option `-clus` lists all service groups on the cluster designated by the variable *cluster*; the option `-localclus` specifies the local cluster.

To display usage for the service group command

- ◆ Use the following command to display usage for the service group command:

```
hagrp [-help [-modify|-link|-list]]
```

Querying resources across clusters

This topic describes how to perform queries on resources:

To display resource attribute values across clusters

- ◆ Use the following command to display resource attribute values across clusters:

```
hares -value resource attribute [system]  
[-clus cluster | -localclus]
```

The option `-clus` displays the attribute value on the cluster designated by the variable `cluster`; the option `-localclus` specifies the local cluster.

If the attribute has local scope, you must specify the system name, except when querying the attribute on the system from which you run the command.

To display the state of a resource across clusters

- ◆ Use the following command to display the state of a resource across clusters:

```
hares -state [resource -sys system]  
[-clus cluster | -localclus]
```

The option `-clus` displays the state of all resources on the specified cluster; the option `-localclus` specifies the local cluster. Specifying a system displays resource state on a particular system.

To display resource information across clusters

- ◆ Use the following command to display resource information across clusters:

```
hares -display [resources] [-attribute attributes]  
[-group service_groups] [-type types] [-sys systems]  
[-clus cluster | -localclus]
```

The option `-clus` lists all service groups on the cluster designated by the variable `cluster`; the option `-localclus` specifies the local cluster.

To display a list of resources across clusters

- ◆ Use the following command to display a list of resources across clusters:

```
hares -list [conditionals] [-clus cluster | -localclus]
```

The option `-clus` lists all resources that meet the specified conditions in global service groups on a cluster as designated by the variable `cluster`.

To display usage for the resource command

- ◆ Use the following command to display usage for the resource command:

```
hares -help [-modify | -list]
```

Querying systems

This topic describes how to perform queries on systems:

To display system attribute values across clusters

- ◆ Use the following command to display system attribute values across clusters:

```
hasys -value system attribute [-clus cluster | -localclus]
```

The option `-clus` displays the values of a system attribute in the cluster as designated by the variable *cluster*; the option `-localclus` specifies the local cluster.

To display the state of a system across clusters

- ◆ Use the following command to display the state of a system across clusters:

```
hasys -state [system] [-clus cluster | -localclus]
```

Displays the current state of the specified system. The option `-clus` displays the state in a cluster designated by the variable *cluster*; the option `-localclus` specifies the local cluster. If you do not specify a system, the command displays the states of all systems.

For information about each system across clusters

- ◆ Use the following command to display information about each system across clusters:

```
hasys -display [systems] [-attribute attributes] [-clus cluster | -localclus]
```

The option `-clus` displays the attribute values on systems (if specified) in a cluster designated by the variable *cluster*; the option `-localclus` specifies the local cluster.

For a list of systems across clusters

- ◆ Use the following command to display a list of systems across clusters:

```
hasys -list [conditionals] [-clus cluster | -localclus]
```

Displays a list of systems whose values match the given conditional statements. The option `-clus` displays the systems in a cluster designated by the variable *cluster*; the option `-localclus` specifies the local cluster.

Querying clusters

This topic describes how to perform queries on clusters:

For the value of a specific cluster attribute on a specific cluster

- ◆ Use the following command to obtain the value of a specific cluster attribute on a specific cluster:

```
haclus -value attribute [cluster] [-localclus]
```

The attribute must be specified in this command. If you do not specify the cluster name, the command displays the attribute value on the local cluster.

To display the state of a local or remote cluster

- ◆ Use the following command to display the state of a local or remote cluster:

```
haclus -state [cluster] [-localclus]
```

The variable *cluster* represents the cluster. If a cluster is not specified, the state of the local cluster and the state of all remote cluster objects as seen by the local cluster are displayed.

For information on the state of a local or remote cluster

- ◆ Use the following command for information on the state of a local or remote cluster:

```
haclus -display [cluster] [-localclus]
```

If a cluster is not specified, information on the local cluster is displayed.

For a list of local and remote clusters

- ◆ Use the following command for a list of local and remote clusters:

```
haclus -list [conditionals]
```

Lists the clusters that meet the specified conditions, beginning with the local cluster.

To display usage for the cluster command

- ◆ Use the following command to display usage for the cluster command:

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

To display the status of a faulted cluster

- ◆ Use the following command to display the status of a faulted cluster:

```
haclus -status cluster
```

Displays the status on the specified faulted cluster. If no cluster is specified, the command displays the status on all faulted clusters. It lists the service groups that were not in the OFFLINE or the FAULTED state before the fault occurred. It also suggests corrective action for the listed clusters and service groups.

Querying status

This topic describes how to perform queries on status of remote and local clusters:

For the status of local and remote clusters

- ◆ Use the following command to obtain the status of local and remote clusters:

```
hastatus
```

Querying heartbeats

The hahb command is used to manage WAN heartbeats that emanate from the local cluster. Administrators can monitor the "health of the remote cluster via heartbeat commands and mechanisms such as Internet, satellites, or storage replication technologies. Heartbeat commands are applicable only on the cluster from which they are issued.

Note: You must have Cluster Administrator privileges to add, delete, and modify heartbeats.

The following commands are issued from the command line.

For a list of heartbeats configured on the local cluster

- ◆ Use the following command for a list of heartbeats configured on the local cluster:

```
hahb -list [conditionals]
```

The variable *conditionals* represents the conditions that must be met for the heartbeat to be listed.

To display information on heartbeats configured in the local cluster

- ◆ Use the following command to display information on heartbeats configured in the local cluster:

```
hahb -display [heartbeat ...]
```

If *heartbeat* is not specified, information regarding all heartbeats configured on the local cluster is displayed.

To display the state of the heartbeats in remote clusters

- ◆ Use the following command to display the state of heartbeats in remote clusters:

```
hahb -state [heartbeat] [-clus cluster]
```

For example, to get the state of heartbeat *Icmp* from the local cluster to the remote cluster *phoenix*:

```
hahb -state Icmp -clus phoenix
```

To display an attribute value of a configured heartbeat

- ◆ Use the following command to display an attribute value of a configured heartbeat:

```
hahb -value heartbeat attribute [-clus cluster]
```

The `-value` option provides the value of a single attribute for a specific heartbeat. The cluster name must be specified for cluster-specific attribute values, but not for global.

For example, to display the value of the *ClusterList* attribute for heartbeat *Icmp*:

```
hahb -value Icmp ClusterList
```

Note that *ClusterList* is a global attribute.

To display usage for the command `hahb`

- ◆ Use the following command to display usage for the command `hahb`:

```
hahb [-help [-modify]]
```

If the `-modify` option is specified, the usage for the `hahb -modify` option is displayed.

Administering global service groups in a global cluster setup

Operations for the VCS global clusters option are enabled or restricted depending on the permissions with which you log on. The privileges associated with each user role are enforced for cross-cluster, service group operations.

This topic includes commands to administer global service groups.

See the `hagrp` (1M) manual page for more information.

To administer global service groups in a global cluster setup

- ◆ Depending on the administrative task you want to perform on global service groups, run the `hagrp` command as follows:

To bring a service group online across clusters for the first time

```
hagrp -online -force
```

To bring a service group online across clusters

```
hagrp -online service_group -sys system [-clus cluster | -localclus]
```

The option `-clus` brings the service group online on the system designated in the cluster. If a system is not specified, the service group is brought online on any node within the cluster. The option `-localclus` brings the service group online in the local cluster.

To bring a service group online on any node

```
hagrp -online [-force] service_group -any [-clus cluster | -localclus]
```

The option `-any` specifies that HAD brings a failover group online on the optimal system, based on the requirements of service group workload management and existing group dependencies. If bringing a parallel group online, HAD brings the group online on each system designated in the `SystemList` attribute.

To display the resources for a service group

```
hagrp -resources service_group [-clus cluster_name | -localclus]
```

The option `-clus` displays information for the cluster designated by the variable `cluster_name`; the option `-localclus` specifies the local cluster.

To take a service group offline across clusters

```
hagrp -offline [-force] [-ifprobed] service_group  
-sys system [-clus cluster | -localclus]
```

The option `-clus` takes offline the service group on the system designated in the cluster.

To take a service group offline anywhere

```
hagrp -offline [-ifprobed] service_group -any  
[-clus cluster | -localclus]
```

The option `-any` specifies that HAD takes a failover group offline on the system on which it is online. For a parallel group, HAD takes the group offline on each system on which the group is online. HAD adheres to the existing group dependencies when taking groups offline.

To switch a service group across clusters

```
hagrp -switch service_group -to system [-clus  
cluster | -localclus [-nopre]]
```

The option `-clus` identifies the cluster to which the service group will be switched. The service group is brought online on the system specified by the `-to system` argument. If a system is not specified, the service group may be switched to any node within the specified cluster.

The option `-nopre` indicates that the VCS engine must switch the service group regardless of the value of the PreSwitch service group attribute.

To switch a service group anywhere

```
hagrp -switch service_group -any [-clus cluster  
| -localclus]
```

The `-any` option specifies that the VCS engine switches a service group to the best possible system on which it is currently not online, based on the value of the group's FailOverPolicy attribute. The VCS engine switches a global service group from a system to another system in the local cluster or a remote cluster.

If you do not specify the `-clus` option, the VCS engine by default assumes `-localclus` option and selects an available system within the local cluster.

The option `-clus` identifies the remote cluster to which the service group will be switched. The VCS engine then selects the target system on which to switch the service group.

To switch a parallel global service group across clusters

```
hagrp -switch
```

VCS brings the parallel service group online on all possible nodes in the remote cluster.

Administering resources in a global cluster setup

This topic describes how to administer resources.

See the `hares` (1M) manual page for more information.

To administer resources in a global cluster setup

- ◆ Depending on the administrative task you want to perform for resources, run the `hares` command as follows:

To take action on a resource across clusters

```
hares -action resource token [-actionargs arg1 ...] [-sys system] [-clus cluster | -localclus]
```

The option `-clus` implies resources on the cluster. If the designated system is not part of the local cluster, an error is displayed. If the `-sys` option is not used, it implies resources on the local node.

To invoke the Info function across clusters

```
hares -refreshinfo resource [-sys system] [-clus cluster | -localclus]
```

Causes the Info function to update the value of the ResourceInfo resource level attribute for the specified resource if the resource is online. If no system or remote cluster is specified, the Info function runs on local system(s) where the resource is online.

To display usage for the resource command

```
hares [-help [-modify | -list]]
```

Administering clusters in global cluster setup

The topic includes commands that are used to administer clusters in a global cluster setup.

See the `haclus` (1M) manual page for more information.

To administer clusters in global cluster setup

- ◆ Depending on the administrative task you want to perform on the clusters, run the `haclus` command as follows:

The variable `cluster` in the following commands represents the cluster.

To add a remote cluster object

```
haclus -add cluster ip
```

This command does not apply to the local cluster.

To delete a remote cluster object	<code>haclus -delete cluster</code>
To modify an attribute of a local or remote cluster object	<code>haclus -modify attribute value [-clus cluster]...</code>
To declare the state of a cluster after a disaster	<code>haclus -declare disconnnet/outage/disaster/replica -clus cluster [-failover]</code>
To manage cluster alerts	See “Managing cluster alerts in a global cluster setup” on page 443.
To change the cluster name	See “Changing the cluster name in a global cluster setup” on page 444.

Managing cluster alerts in a global cluster setup

This topic includes commands to manage cluster alerts.

See the `haalert` (1M) manual page for more information.

To manage cluster alerts

- ◆ Run the `haalert` command to manage cluster alerts.

<code>haalert -testfd</code>	Generates a simulated "cluster fault" alert that is sent to the VCS engine and GUI.
<code>haalert -display</code>	For each alert, the command displays the following information: <ul style="list-style-type: none"> ■ alert ID ■ time when alert occurred ■ cluster on which alert occurred ■ object name for which alert occurred ■ (cluster name, group name, and so on). ■ informative message about alert
<code>haalert -list</code>	For each alert, the command displays the following information: <ul style="list-style-type: none"> ■ time when alert occurred ■ alert ID

```
haalert -delete  
alert_id -notes  
"description"
```

Deletes a specific alert. You must enter a text message within quotes describing the reason for deleting the alert. The comment is written to the engine log as well as sent to any connected GUI clients.

```
haalert -help
```

Displays the usage text

Changing the cluster name in a global cluster setup

This topic describes how to change the `ClusterName` attribute in a global cluster configuration. The instructions describe how to rename `VCSPriCluster` to `VCSPriCluster2` in a two-cluster configuration, comprising clusters `VCSPriCluster` and `VCSecCluster` configured with the global group `AppGroup`.

Before changing the cluster name, make sure the cluster is not part of any `ClusterList`, in the wide-area Heartbeat agent and in global service groups.

To change the name of a cluster

- 1 Run the following commands from cluster `VCSPriCluster`:

```
hagrp -offline ClusterService -any  
hagrp -modify AppGroup ClusterList -delete VCSPriCluster  
haclus -modify ClusterName VCSPriCluster2  
hagrp -modify AppGroup ClusterList -add VCSPriCluster2 0
```

- 2 Run the following commands from cluster `VCSecCluster`:

```
hagrp -offline ClusterService -any  
hagrp -modify appgrp ClusterList -add VCSPriCluster  
hahb -modify Icmp ClusterList -delete VCSPriCluster  
haclus -delete VCSPriCluster  
haclus -add VCSPriCluster2 your_ip_address  
hahb -modify Icmp ClusterList -add VCSPriCluster2  
hahb -modify Icmp Arguments your_ip_address -clus VCSPriCluster2  
hagrp -modify AppGroup ClusterList -add VCSPriCluster2 0  
hagrp -online ClusterService -any
```

- 3 Run the following command from the cluster renamed to `VCSPriCluster2`:

```
hagrp -online ClusterService -any
```

Administering heartbeats in a global cluster setup

This topic includes commands that are used to administer heartbeats.

See the `hahb` (1M) manual page for more information.

To administer heartbeats in a global cluster setup

- ◆ Depending on the administrative task you want to perform for heartbeats, run the `hahb` command as follows:

To create a heartbeat

```
hahb -add heartbeat
```

For example, type the following command to add a new `IcmpS` heartbeat. This represents a heartbeat sent from the local cluster and immediately forks off the specified agent process on the local cluster.

```
hahb -add IcmpS
```

To modify a heartbeat

```
hahb -modify heartbeat attribute value ... [-clus cluster]
```

If the attribute is local, that is, it has a separate value for each remote cluster in the `ClusterList` attribute, the option `-clus cluster` must be specified. Use `-delete -keys` to clear the value of any list attributes.

For example, type the following command to modify the `ClusterList` attribute and specify targets "phoenix" and "houston" for the newly created heartbeat:

```
hahb -modify Icmp ClusterList phoenix houston
```

To modify the `Arguments` attribute for target phoenix:

```
hahb -modify Icmp Arguments phoenix.example.com  
-clus phoenix
```

To delete a heartbeat

```
hahb -delete heartbeat
```

To change the scope of an attribute to cluster-specific

```
hahb -local heartbeat attribute
```

For example, type the following command to change the scope of the attribute `AYAInterval` from global to cluster-specific:

```
hahb -local Icmp AYAInterval
```

To change the
scope of an
attribute to global

```
hahb -global heartbeat attribute value ... | key  

... | key value ...
```

For example, type the following command to change the scope of
the attribute `AYAInterval` from cluster-specific to cluster-generic:

```
hahb -global Icmp AYAInterval 60
```

Setting up replicated data clusters

This chapter includes the following topics:

- [About replicated data clusters](#)
- [How VCS replicated data clusters work](#)
- [About setting up a replicated data cluster configuration](#)

About replicated data clusters

The Replicated Data Cluster (RDC) configuration provides both local high availability and disaster recovery functionality in a single VCS cluster.

You can set up RDC in a VCS environment using Volume Replicator (VVR).

A Replicated Data Cluster (RDC) uses data replication to assure data access to nodes. An RDC exists within a single VCS cluster. In an RDC configuration, if an application or a system fails, the application is failed over to another system within the current primary site. If the entire primary site fails, the application is migrated to a system in the remote secondary site (which then becomes the new primary).

For VVR replication to occur, the disk groups containing the Replicated Volume Group (RVG) must be imported at the primary and secondary sites. The replication service group must be online at both sites simultaneously, and must be configured as a hybrid VCS service group.

The application service group is configured as a failover service group. The application service group must be configured with an *online local hard* dependency on the replication service group.

Note: VVR supports multiple replication secondary targets for any given primary. However, RDC for VCS supports only one replication secondary for a primary.

An RDC configuration is appropriate in situations where dual dedicated LLT links are available between the primary site and the disaster recovery secondary site but lacks shared storage or SAN interconnect between the primary and secondary data centers. In an RDC, data replication technology is employed to provide node access to data in a remote site.

Note: You must use dual dedicated LLT links between the replicated nodes.

How VCS replicated data clusters work

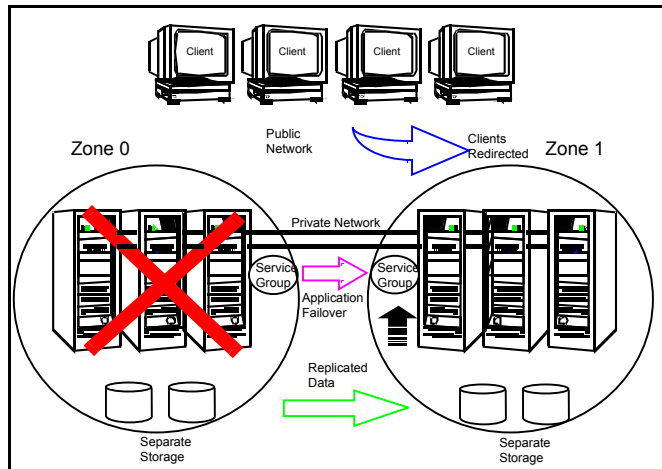
To understand how a replicated data cluster configuration works, let us take the example of an application configured in a VCS replicated data cluster. The configuration has two system zones:

- Primary zone (zone 0) comprising nodes located at the primary site and attached to the primary storage
- Secondary zone (zone 1) comprising nodes located at the secondary site and attached to the secondary storage

The application is installed and configured on all nodes in the cluster. Application data is located on shared disks within each RDC site and is replicated across RDC site to ensure data concurrency. The application service group is online on a system in the current primary zone and is configured to fail over in the cluster.

Figure 16-1 depicts an application configured on a VCS replicated data cluster.

Figure 16-1 A VCS replicated data cluster configuration



In the event of a system or application failure, VCS attempts to fail over the application service group to another system within the same RDC site. However, in the event that VCS fails to find a failover target node within the primary RDC site, VCS switches the service group to a node in the current secondary RDC site (zone 1).

About setting up a replicated data cluster configuration

Depending on your application, refer to one of the following solutions guides for detailed configuration information:

- *Storage Foundation and High Availability Solutions HA and Disaster Recovery Solutions Guide for Microsoft Exchange 2010*
- *Storage Foundation and High Availability Solutions HA and Disaster Recovery Solutions Guide for Microsoft SQL Server*
- *Storage Foundation and High Availability Solutions HA and Disaster Recovery Solutions Guide*

Troubleshooting and performance

- [Chapter 17. VCS performance considerations](#)
- [Chapter 18. Troubleshooting and recovery for VCS](#)

VCS performance considerations

This chapter includes the following topics:

- [How cluster components affect performance](#)
- [How cluster operations affect performance](#)
- [Monitoring CPU usage](#)
- [VCS agent statistics](#)
- [About VCS performance with non-HA products](#)
- [About VCS performance with SFW](#)

How cluster components affect 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 main components of clustering that have an impact on performance include the kernel; specifically, GAB and LLT, the VCS engine (HAD), and the VCS agents. 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.

See [“How kernel components \(GAB and LLT\) affect performance”](#) on page 452.

See [“How the VCS engine \(HAD\) affects performance”](#) on page 452.

See [“How agents affect performance”](#) on page 453.

See [“How the VCS graphical user interfaces affect performance”](#) on page 454.

How kernel components (GAB and LLT) affect performance

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 `%VCS_HOME%\comms\llt\llttab.txt` configuration file.

System-to-system communication is load-balanced across all private network links. If a link fails, VCS continues to use all remaining links. 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, which by default generates a small (approximately 64-byte) broadcast packet per second from each system, and which will carry data only when all private network links have failed.

How the VCS engine (HAD) affects performance

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. HAD runs logging activities in a separate thread to reduce the performance impact on the engine due to logging.

VCS runs 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 involve taking the group offline on one system and bringing it online on another system. A system fault invokes failing over all online service groups on the faulted system.

To continuously monitor VCS status, use the VCS graphical user interfaces 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.

How agents affect performance

The VCS agent processes have the most impact on system performance. Each agent process has two components: the agent framework and the agent functions. The agent framework provides common functionality, such as communication with the HAD, multithreading for multiple resources, scheduling threads, and invoking functions. Agent functions implement agent-specific functionality. Review the performance guidelines to follow when configuring agents.

See [“Monitoring resource type and agent configuration”](#) on page 453.

Monitoring 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 resource 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. 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 30.

Continuing with this example, the Mount agent schedules the `monitor` function 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` function 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` function 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 functions by modifying their associated attributes. The attributes `MonitorTimeout`, `OnlineTimeout`, and `OfflineTimeout` indicate the maximum time (in seconds) within which the monitor, online, and offline functions 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, Veritas recommends measuring the time it takes to bring a resource online, take it offline, and monitor before modifying the defaults. 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 and 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 function 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 is proving costly, you can divide it into cursory, or shallow monitoring, and the more extensive deep (or in-depth) monitoring. Whether to use shallow or deep monitoring depends on your configuration requirements.

As an additional consideration for agents, properly configure the attribute `SystemList` for your service group. For example, if you know that a service group can go online on `SystemA` and `SystemB` only, do not include other systems in the `SystemList`. This saves additional agent processes and monitoring overhead.

How the VCS graphical user interfaces affect performance

The VCS graphical user interface, Cluster Manager (Java Console) maintains a persistent connection to HAD, from which it receives regular updates regarding cluster status. For best results, run the GUI on a system outside the cluster to avoid impact on node performance.

If the network adapters cannot ping each other, the cluster nodes may not get GAB membership

While configuring LLT over UDP, if the network adapters selected for the LLT communication cannot ping each other and you proceed with the cluster configuration, VCS configures the LLT service on the selected network adapters but the cluster nodes may not receive GAB membership and as a result the Veritas High Availability Engine, HAD, may fail to start.

You can confirm the GAB membership by running the following command:

```
gabconfig -a
```

If no port membership information is returned it indicates that GAB is not operating. This issue can be addressed in either of the following methods:

To address this issue using method 1

- 1 Reboot the cluster nodes that do not have GAB membership.
- 2 Verify the GAB operation in the cluster.

Type the following on the command prompt:

```
gabconfig -a
```

If GAB membership information is displayed for all cluster nodes, GAB is working correctly. However, if the appropriate GAB membership information is not returned for one or more nodes, GAB is not operating correctly. In that case, proceed to the next method.

To address this issue using method 2

- 1 Stop the LLT service in the cluster.

Type the following on the command prompt:

```
:net stop llc
```

- 2 Delete the cluster using VCW.
- 3 Ensure that the network adapters can ping each other and then re-create the cluster using VCW.

How cluster operations affect performance

Review the following topics that describe how the following operations on systems, resources, and service groups in the cluster affect performance:

A cluster system boots	See “VCS performance consideration when booting a cluster system” on page 456.
A resource comes online	See “VCS performance consideration when a resource comes online” on page 456.
A resource goes offline	See “VCS performance consideration when a resource goes offline” on page 457.
A service group comes online	See “VCS performance consideration when a service group comes online” on page 457.
A service group goes offline	See “VCS performance consideration when a service group goes offline” on page 458.

A resource fails	See “ VCS performance consideration when a resource fails ” on page 458.
A system fails	See “ VCS performance consideration when a system fails ” on page 459.
A network link fails	See “ VCS performance consideration when a network link fails ” on page 459.
A system panics	See “ VCS performance consideration when a system panics ” on page 460.
A service group switches over	See “ VCS performance consideration when a service group switches over ” on page 462.
A service group fails over	See “ VCS performance consideration when a service group fails over ” on page 462.

VCS performance consideration when 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. After building the configuration database, the system 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: Bringing service groups online as part of AutoStart occurs after VCS transitions to `RUNNING` mode.

VCS performance consideration when a resource comes online

The `online` function of an agent brings the resource online. This function 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 function finishes. The agent schedules the monitor immediately after the function 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.

VCS performance consideration when a resource goes offline

Similar to the online function, the offline function 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 function finishes. The agent schedules the monitor immediately after the offline function finishes, so the first monitor detects the resource as offline.

VCS performance consideration when a service group comes 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 online R3. When R3 is online, VCS online R2. When R2 is online, VCS online 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. More complex service group trees do not allow much parallelism and serializes the group online operation.

The time it takes to bring a service group online or take it offline also depends on the type of service group, such as fileshare, or enterprise application agent, etc.

For a fileshare service group, there are four factors that determine how long it takes to bring a fileshare online:

- `ShareSubDirectories`

If set to 1, each child subdirectory is shared. the fileshare group's online entry point shares child folders in addition to parent folders.

- Number of subdirectories
The greater the number of subdirectories being shared, the longer it takes to bring online, monitor, and take offline a fileshare service group.
- Number of permissions
For each share, the online entry point applies the share permissions as configured.
- AutoShare and AutoControl
By default, if ShareSubDirectories is set, the fileshare service group monitors new directories and shares them. AutoShare occurs in the monitor entry points.

VCS performance consideration when a service group goes offline

Taking service groups offline works from the top down, as opposed to the online operation, 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 where R1 depends on R2 and R2 depends on 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.

VCS performance consideration when a resource fails

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` function 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 the ConflInterval attribute, previous reports of offline are not counted against ToleranceLimit.

When the agent determines that the resource is faulted, it calls the clean function (if implemented) to verify that the resource is completely offline. The monitor following clean verifies the offline. The agent then tries to restart the resource according to the number set in the RestartLimit attribute (if the value of the attribute is non-zero) before it gives up and informs HAD that the resource is faulted. However, if the resource remains online for the interval designated in ConflInterval, 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 function is implemented. Therefore, the time it takes to detect

failure depends on the MonitorInterval, the efficiency of the monitor and clean (if implemented) functions, and the ToleranceLimit (if set).

VCS performance consideration when a system fails

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 `%VCS_HOME%\comms\llt\llttab.txt` file.

For example, to specify 12 seconds:

```
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 boots, 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. You can modify the default value of the ShutdownTimeout attribute.

See [System attributes](#) on page 570.

VCS performance consideration when a network link fails

If a system loses a network link to the cluster, other systems stop receiving heartbeats over the links from that system. LLT detects this and waits for 16 seconds before declaring the system lost a link.

See “[VCS performance consideration when a system fails](#)” on page 459.

You can modify the LLT peer inactive timeout value in the `%VCS_HOME%\comms\llt\llttab.txt` file.

For example, to specify 12 seconds:

```
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`.

VCS performance consideration when a system panics

There are several instances in which GAB will intentionally panic a system. For example, GAB panics a system if it detects an internal protocol error or discovers an LLT node-ID conflict. Other instances are as follows:

- Client process failure
See [“About GAB client process failure”](#) on page 460.
- Registration monitoring
See [“About GAB client registration monitoring”](#) on page 461.
- Network failure
See [“About network failure and GAB IOFENCE message”](#) on page 462.
- Quick reopen
See [“About quick reopen”](#) on page 462.

About GAB client process failure

If a GAB client process such as HAD 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 effect on the entire cluster. If the `-b` option is used in `gabconfig`, GAB does not try to kill the client process. Instead, it panics the system when the client process fails to heartbeat. This option cannot be turned off once set.

By default, GAB tries to kill HAD five times before closing the port. The number of times GAB tries to kill HAD is a kernel tunable parameter, `gab_kill_ntries`, and is configurable. The minimum value for this tunable is 3 and the maximum is 10.

Port closure is an indication to other nodes that HAD on this node has been killed. Should HAD recover from its stuck state, it first processes pending signals. Here it receives the SIGKILL first and gets killed.

After GAB sends a SIGKILL signal, it waits for a specific amount of time for HAD to get killed. If HAD survives beyond this time limit, GAB panics the system. This time limit is a kernel tunable parameter, `gab_isolate_time`, and is configurable. The minimum value for this timer is 16 seconds and maximum is 4 minutes.

About GAB client registration monitoring

The registration monitoring feature lets you configure GAB behavior when HAD is killed and does not reconnect after a specified time interval.

This scenario may occur in the following situations:

- The system is very busy and the hashadow process cannot restart HAD.
- The HAD and hashadow processes were killed by user intervention.
- The hashadow process restarted HAD, but HAD could not register.
- A hardware failure causes termination of the HAD and hashadow processes.
- Any other situation where the HAD and hashadow processes are not run.

When this occurs, the registration monitoring timer starts. GAB takes action if HAD does not register within the time defined by the `VCS_GAB_RMTIMEOUT` parameter, which is defined in the `vcseenv` file. The default value for `VCS_GAB_RMTIMEOUT` is 200 seconds.

When HAD cannot register after the specified time period, GAB logs a message every 15 seconds saying it will panic the system.

You can control GAB behavior in this situation by setting the `VCS_GAB_RMACTION` parameter in the `vcseenv` file.

- To configure GAB to panic the system in this situation, set:

```
VCS_GAB_RMACTION=panic
```

In this configuration, killing the HAD and hashadow processes results in a panic unless you start HAD within the registration monitoring timeout interval.

- To configure GAB to log a message in this situation, set:

```
VCS_GAB_RMACTION=SYSLOG
```

The default value of this parameter is `SYSLOG`, which configures GAB to log a message when HAD does not reconnect after the specified time interval.

In this scenario, you can choose to restart HAD (using `hastart`) or restart the GAB service.

When you enable registration monitoring, GAB takes no action if the HAD process unregisters with GAB normally, that is if you stop HAD using the `hastop` command.

About network failure and GAB IOFENCE message

If a network partition occurs, a cluster can split into two or more separate sub-clusters. When two clusters join as one, GAB ejects one sub-cluster. GAB prints diagnostic messages and sends iofence messages to the sub-cluster being ejected.

The systems in the sub-cluster process the iofence messages depending on the type of GAB port that a user client process or a kernel module uses:

- If the GAB client is a user process, then GAB tries to kill the client process.
- If the GAB client is a kernel module, then GAB panics the system.

The `gabconfig` command's `-k` and `-j` options apply to the user client processes. The `-k` option prevents GAB from panicking the system when it cannot kill the user processes. The `-j` option panics the system and does not kill the user process when GAB receives the iofence message.

About 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.

VCS performance consideration when a service group switches over

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.

VCS performance consideration when a service group fails over

The time it takes to fail over a service group when a resource faults equals the following:

- 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 fail over a service group when a system faults equals the following:

- The time it takes to detect system fault

- The time it takes to offline the dependent service groups on other running systems
- 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.

Monitoring CPU usage

VCS includes a system attribute, `CPUUsageMonitoring`, which monitors CPU usage on a specific system and notifies the administrator when usage has been exceeded.

The default values for the `CPUUsageMonitoring` attribute are:

- `Enabled` = 0
- `NotifyThreshold` = 0
- `NotifyTimeLimit` = 0
- `ActionThreshold` = 0
- `ActionTimeLimit` = 0
- `Action` = NONE.

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.

If `Enabled` is set to 1, HAD monitors the usage and updates `CPUUsage` attribute. If `Enabled` is set to 0 (default), HAD does not monitor the usage.

If the system's CPU usage continuously exceeds the value set in `NotifyThreshold` for a duration greater than the value set in `NotifyTimeLimit`, HAD sends notification via an SNMP trap or SMTP message.

If the CPU usage continuously exceeds the value set in `NotifyThreshold` for a duration greater than the value set in `NotifyTimeLimit`, subsequent notifications are sent after five minutes to avoid sending notifications too frequently (if the `NotifyTimeLimit` value is set to a value less than five minutes). In this case, notification is sent after the first interval of `NotifyTimeLimit`. As CPU usage continues

to exceed the threshold value, notifications are sent after five minutes. If the values of `NotifyThreshold` or `NotifyTimeLimit` are set to 0, no notification is sent.

If system's CPU usage exceeds the value set in `ActionThreshold` continuously for a duration greater than the value set in `ActionTimeLimit`, the specified action is taken. If the CPU usage continuously exceeds the `ActionThreshold` for a duration greater than the value set in `ActionTimeLimit`, subsequent action is taken after five minutes to avoid taking action too frequently (if the `ActionTimeLimit` value is set to less than five minutes). In this case action is taken after the first interval of `ActionTimeLimit`. As CPU usage continues to exceed the threshold value, action is taken after five minutes. If the values of `ActionThreshold` or `ActionTimeLimit` are set to 0, no action is taken. Actions can have one of the following values:

NONE: No action will be taken and the message is logged in the VCS engine log.

REBOOT: System is rebooted.

CUSTOM: The `cpuusage` trigger is invoked.

VCS agent statistics

You can configure VCS to track the time taken for monitoring resources.

You can also detect potential problems with resources and systems on which resources are online by analyzing the trends in the time taken by the resource's monitor cycle. Note that VCS keeps track of monitor cycle times for online resources only.

VCS calculates the time taken for a monitor cycle to complete and computes an average of monitor times after a specific number of monitor cycles and stores the average in a resource-level attribute.

VCS also tracks increasing trends in the monitor cycle times and sends notifications about sudden and gradual increases in monitor times.

VCS uses the following parameters to compute the average monitor time and to detect increasing trends in monitor cycle times:

- **Frequency:** The number of monitor cycles after which the monitor time average is computed and sent to the VCS engine.
For example, if `Frequency` is set to 10, VCS computes the average monitor time after every 10 monitor cycles.
- **ExpectedValue:** The expected monitor time (in milliseconds) for a resource.
VCS sends a notification if the actual monitor time exceeds the expected monitor time by the `ValueThreshold`. So, if you set this attribute to 5000 for a `FileOnOff` resource, and if `ValueThreshold` is set to 40%, VCS will send a notification only

when the monitor cycle for the FileOnOff resource exceeds the expected time by over 40%, that is 7000 milliseconds.

- **ValueThreshold:** The maximum permissible deviation (in percent) from the expected monitor time. When the time for a monitor cycle exceeds this limit, VCS sends a notification about the sudden increase or decrease in monitor time.

For example, a value of 100 means that VCS sends a notification if the actual monitor time deviates from the expected time by over 100%.

VCS sends these notifications conservatively. If 12 consecutive monitor cycles exceed the threshold limit, VCS sends a notification for the first spike, and then a collective notification for the next 10 consecutive spikes.

- **AvgThreshold:** The threshold value (in percent) for increase in the average monitor cycle time for a resource.

VCS maintains a running average of the time taken by the monitor cycles of a resource. The first such computed running average is used as a benchmark average. If the current running average for a resource differs from the benchmark average by more than this threshold value, VCS regards this as a sign of gradual increase or decrease in monitor cycle times and sends a notification about it for the resource. Whenever such an event occurs, VCS resets the internally maintained benchmark average to this new average. VCS sends notifications regardless of whether the deviation is an increase or decrease in the monitor cycle time.

For example, a value of 25 means that if the actual average monitor time is 25% more than the benchmark monitor time average, VCS sends a notification.

Tracking monitor cycle times

VCS marks sudden changes in monitor times by comparing the time taken for each monitor cycle with the ExpectedValue. If this difference exceeds the ValueThreshold, VCS sends a notification about the sudden change in monitor time. Note that VCS sends this notification only if monitor time increases.

VCS marks gradual changes in monitor times by comparing the benchmark average and the moving average of monitor cycle times. VCS computes the benchmark average after a certain number of monitor cycles and computes the moving average after every monitor cycle. If the current moving average exceeds the benchmark average by more than the AvgThreshold, VCS sends a notification about this gradual change in the monitor cycle time.

VCS attributes enabling agent statistics

This topic describes the attributes that enable VCS agent statistics.

MonitorStatsParam A resource type-level attribute, which stores the required parameter values for calculating monitor time statistics.

```
static str MonitorStatsParam = { Frequency = 10,
ExpectedValue = 3000, ValueThreshold = 100,
AvgThreshold = 40 }
```

- **Frequency:** Defines the number of monitor cycles after which the average monitor cycle time should be computed and sent to the engine. If configured, the value for this attribute must be between 1 and 30. It is set to 0 by default.
- **ExpectedValue:** The expected monitor time in milliseconds for all resources of this type. Default=3000.
- **ValueThreshold:** The acceptable percentage difference between the expected monitor cycle time (ExpectedValue) and the actual monitor cycle time. Default=100.
- **AvgThreshold:** The acceptable percentage difference between the benchmark average and the moving average of monitor cycle times. Default=40

MonitorTimeStats Stores the average time taken by a number of monitor cycles specified by the Frequency attribute along with a timestamp value of when the average was computed.

```
str MonitorTimeStats{} = { Avg = "0", TS = "" }
```

This attribute is updated periodically after a number of monitor cycles specified by the Frequency attribute. If Frequency is set to 10, the attribute stores the average of 10 monitor cycle times and is updated after every 10 monitor cycles.

The default value for this attribute is 0.

ComputeStats A flag that specifies whether VCS keeps track of the monitor times for the resource.

```
boolean ComputeStats = 0
```

The value 0 indicates that VCS will not keep track of the time taken by the monitor routine for the resource. The value 1 indicates that VCS keeps track of the monitor time for the resource.

The default value for this attribute is 0.

About VCS performance with non-HA products

To ensure optimum performance, it is important to evaluate the impact of non-HA products on cluster nodes. Evaluating factors such as the complexity of the VCS

configuration, the capacity of the hardware to host multiple applications, and the intended use of the product will assist you in determining how and where to host the applications.

When modifying the system, consider whether or not the change will cause the service group to fault. A simple task such as Windows Explorer browsing fileshares hosted by VCS may seem harmless, but it would prevent VCS from failing over because the drive is locked by another application.

About VCS performance with SFW

If you use Veritas Storage Foundation for Windows (SFW) on clustered nodes, we strongly recommend the following:

- Carefully evaluate changes to underlying storage. Typically, changes to the volume and disk group configurations require corresponding changes to the VCS configuration. Common changes include unassigning or reassigning the drive letters, splitting or joining a disk group, or snapshotting the volume. Prior to implementing these types of changes, evaluate your configuration to determine whether to freeze, offline, or fail over the VCS service groups to avoid faulting the groups inadvertently.
- Like Cluster Manager, the SFW GUI runs under the Java Runtime environment and maintains a persistent connection to the SFW engine, from which it receives regular updates regarding status. For best results, run the SFW GUI on a system outside the cluster. This will avoid potential impact on node performance.
- Certain SFW operations, such as rescan, resync, etc., are CPU-intensive and can affect VCS performance. The VCS kernel module GAB expects the VCS engine, HAD, to send heartbeats that ensure the engine is functioning properly. If the heartbeat interval exceeds five seconds the engine logs an error.

By default, if GAB does not receive a heartbeat from HAD within 15 seconds, GAB assumes something is wrong and kills HAD (which then gets restarted by hashadow). You can tune this interval by changing the value of the system variable `VCS_GAB_TIMEOUT`, which specifies the number of seconds GAB waits for a heartbeat before killing HAD.

Troubleshooting and recovery for VCS

This chapter includes the following topics:

- [VCS message logging](#)
- [Handling network failure](#)
- [Troubleshooting VCS startup](#)
- [Troubleshooting secure clusters](#)
- [Troubleshooting service groups](#)
- [Troubleshooting resources](#)
- [Troubleshooting notification](#)
- [Troubleshooting and recovery for global clusters](#)
- [Troubleshooting the steward process](#)
- [VCS utilities](#)

VCS message logging

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 `%VCS_HOME%\log\engine_A.txt`. The format of engine log messages is:

```
Timestamp (Year/MM/DD) | Mnemonic | Severity | UMI | Message Text
```

The agent log components are defined as follows:

- Timestamp: the date and time the message was generated.
- Mnemonic: the string ID that represents the product (for example, VCS).
- Severity: levels include CRITICAL, ERROR, WARNING, NOTICE, and INFO (most to least severe, respectively).
- UMI: a unique message ID.
- Message Text: the actual message generated by VCS.

A typical engine log resembles:

```
2011/02/10 16:08:09 VCS INFO V-16-1-10077 received new cluster
membership.
```

The agent log is located at %VCS_HOME%\log\agent_A.txt. The format of agent log messages is:

```
Timestamp (Year/MM/DD) | Mnemonic | Severity | UMI | Agent Type |
Resource Name | Entry Point | Message Text
```

A typical agent log resembles:

```
2011/02/23 10:38:23 VCS WARNING V-16-2-23331
Oracle:VRT:monitor:Open for ora_lgwr failed, setting cookie to
null.
```

VCW logs

The VCS Cluster Configuration Wizard (VCW) log is located at

```
%allusersprofile%\Application Data\Veritas\Cluster Server\vcw.log.
```

Here, %allusersprofile% is the file system directory containing application data for all users. A typical path is C:\Documents and Settings\All Users\.

The format of the VCW log is

```
ThreadID | Message Text
```

- ThreadID: the ID of the thread initiated by VCW.
- Message Text: the actual message generated by VCW.

A typical VCW log resembles:

```
00000576-00000264: ExecMethod return 00000000.
00000576-00000110: CRegistry::Query for VCS License failed.
Error=0x00000000
00000576-00000264: ExecMethod return 00000000.
```

```
00000576-00000264: ExecMethod return 00000001.
00000576-00000127: QueryDWORDValue returned 0x00000001
00000576-00000132: CRegistry::Query for VxSS Root information failed.
Error=0x00000001
```

VCWsilent logs

The VCWsilent log is located at <currentdirectory>\vcwsilent.log.

Here, <currentdirectory> is the directory from where the VCWsilent.exe is run.

A typical VCWsilent log resembles:

```
00005540-00000064: 5540: STARTING - Discovering NICs on the
selected machines...
00009956-00000064: 9956: STARTING - Generating private network
related files...
00009956-00000048: 9956: COMPLETED - Generating LLT host
files...
00009956-00000048: 9956: COMPLETED - Generating GAB tab files...
00009956-00000048: 9956: COMPLETED - Generating main.cf file...
00009956-00000064: 9956: STARTING - Configuring LLT on all the nodes.
00009956-00000048: 9956: COMPLETED - Configuring LLT on all the
nodes.
```

Solutions wizard logs

The Solutions Configuration Center (SCC) provides access to many wizards. However, the following three wizards are built in to the Solutions Configuration Center:

- Disaster Recovery Wizard
- Fire Drill Wizard
- Quick Recovery Configuration Wizard

These solutions wizards are launched only from the Solutions Configuration Center, whereas other wizards can be launched from product consoles or the Start menu.

Logs created by these solutions wizards are located in the following paths:

```
C:\ProgramData\Veritas\winsolutions\log
```

Message catalogs

VCS includes multilingual support for message catalogs. Most binary message catalogs (BMCs), are stored in %VCS_HOME%\messages\language\ . The catalogs

`gab.bmc` and `llt.bmc` are stored in `%VCS_ROOT%\comms\messages\language\`. The variable *language* represents a two-letter abbreviation. For example, `en` represents English.

The VCS command-line interface displays error/success messages in any language supported by VCS. The `hamsg` command displays the VCS engine logs in VCS-supported languages.

[Table 18-1](#) shows the complete list of BMCs.

Table 18-1 Binary message catalogs

Module Name	Description
<code>VRTSvcsAgfw.bmc</code>	VCS agent framework messages
<code>VRTSvcsAlerts.bmc</code>	Alerts messages
<code>VRTSvcsApi.bmc</code>	VCS API messages
<code>VRTSvcsCommon.bmc</code>	Common messages
<code>VRTSvcsHad.bmc</code>	VCS engine (HAD) messages
<code>VRTSvcsHbfw.bmc</code>	VCS heartbeat framework messages
<code>VRTSvcsTriggers.bmc</code>	VCS triggers messages
<code>VRTSvcsAgentplatform.bmc</code>	VCS bundled agent messages
<code>VRTSvcsplatformagent_name.bmc</code>	VCS enterprise agent messages
<code>VRTSvcsWac.bmc</code>	Wide-area connector messages
<code>gab.bmc</code>	GAB command-line interface messages
<code>llt.bmc</code>	LLT command-line interface messages

Handling network failure

VCS protects against network partitions by requiring that all systems be connected by two or more communication channels. In a VCS cluster, all systems send heartbeats to each other across communication channels. If a system's heartbeats are not received across one channel, VCS detects that the channel has failed. If a system's heartbeats are not received across any channels, VCS detects that the system has failed. The services running on that system are then restarted on another.

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.

For more information on protecting your cluster against network failure: See [“Verifying LLT, GAB, and cluster operation”](#) on page 480.

Disabling failover

When VCS loses communication with a system, a new regular membership is issued that excludes the departed system. VCS must then determine if it should restart that system's services, or if the system is running services outside of communication with VCS. Two conditions indicate that the system could still be running the services:

- Prior to the system's departure, the systems remaining in the new membership were connected to the departed system by only one communication channel.
- The departed system continues writing heartbeats to disk. VCS detects these conditions using the jeopardy membership.

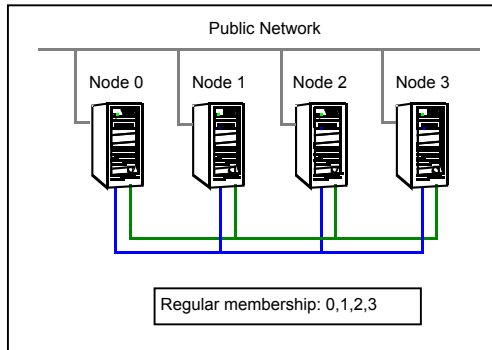
If there is at least one system in the new regular membership that was not part of the prior jeopardy membership, then failover is disabled only for those systems that left the regular membership and were part of the prior jeopardy membership. Failover is also disabled for systems that are in the new jeopardy membership and outside of the new regular membership. This indicates these systems are actively writing heartbeats to disk. If there are no systems in the new regular membership that were not part of the previous jeopardy membership, failover is disabled for all systems that have departed. This indicates that connections from the remaining systems to all systems in the prior regular membership were potentially unreliable.

Example of how VCS handles network failure

In the following example, a single cluster has two networks connecting four nodes.

[Figure 18-1](#) shows an example of a single VCS clusters with four nodes and two networks connecting them.

Figure 18-1 VCS and network failure: Four node cluster

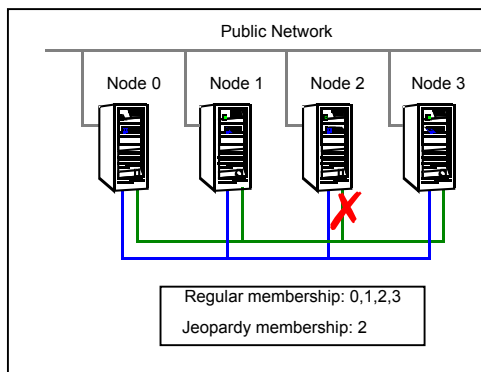


Jeopardy scenario: link failure

In this scenario, a link to node 2 fails, leaving the node with only one possible heartbeat.

Figure 18-2 shows a jeopardy scenario within a four node cluster where a link to node 2 fails.

Figure 18-2 VCS and network failure: Link to node 2 fails.



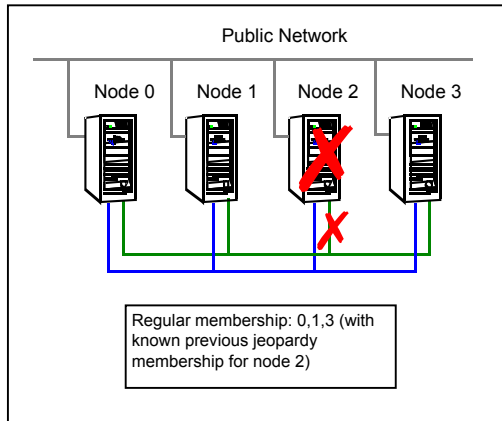
A new cluster membership is issued with nodes 0, 1, 2, and 3 in the regular membership and node 2 in a jeopardy membership. All normal cluster operations continue, including normal failover of service groups due to resource fault.

Jeopardy scenario: link and node failure

Consider that in the previous link-failure scenario, node 2 fails due to a power fault.

Figure 18-3 shows a jeopardy scenario, where node 2 fails and subsequently the service groups running on node 2 also fail leading to link and node failure.

Figure 18-3 VCS and network failure: Node 2 in jeopardy membership

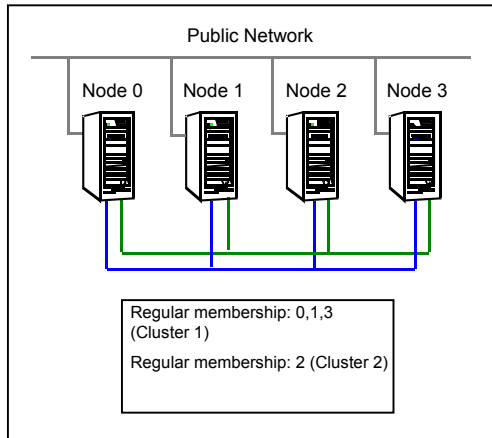


All other systems recognize that node 2 has faulted. In this situation, a new membership is issued for nodes 0, 1 and 3 as regular members. Since node 2 was in a jeopardy membership, service groups running on node 2 are autodisabled, so no other node can assume ownership of these service groups. If the node is actually failed, the system administrator can clear the AutoDisabled flag on the service groups in question and online the groups on other systems in the cluster.

Jeopardy scenario: failure of all links

Figure 18-4 shows a jeopardy scenario where node 2 loses both heartbeats.

Figure 18-4 VCS and network failure: Node 2 forms a single-node-mini cluster



In this situation, a new membership is issued for node 0, 1 and 3 as regular members. Since node 2 was in a jeopardy membership, service groups running on node 2 are autodisabled, so no other node can assume ownership of these service groups. Nodes 0, 1 and 3 form a mini-cluster. Node 2 forms another single-node mini-cluster. All service groups that were present on nodes 0, 1 and 3 are autodisabled on node 2.

Network partitioning

With VCS, two or more communication channels guard against network partitioning; a condition where a failure on the network is misinterpreted as a failure of one or more systems in the cluster. If one system in the cluster assumes wrongly that another system has failed, it may restart applications already running on the other system, thereby corrupting the data.

Using a second communication channel enables VCS to distinguish between network and system failures. If all but one network channel fails, VCS enters a degraded mode that disables automatic application failover caused by system failure. If the last network channel fails, VCS splits into multiple "mini-clusters" without failing over or shutting down applications. This design enables administrative services to operate uninterrupted; for example, you can use VCS to shut down applications during system maintenance. When connections are restored, systems will attempt to rejoin into a single cluster. By default, GAB kills processes associated with ports on rejoining systems. To avoid potential data corruption during rejoin, add the option `-j` to the `gabconfig` command to enable system halt after a split. The `gabconfig` command is located in `%VCS_ROOT%\comms\gab`.

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.

Pre-existing network partitions

A pre-existing 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. VCS seeding is designed to help prevent this situation.

Seeding of VCS clusters

To protect your cluster from a pre-existing network partition, VCS employs the concept of a seed. Nodes can be seeded automatically or manually. Only those nodes that have been seeded can run VCS.

By default, when a node comes up, it is not seeded. When the last node in a cluster is booted, the cluster will seed and start VCS on all the nodes. Nodes can then be brought down and restarted in any combination. Seeding is automatic as long as at least one instance of VCS is running in the cluster.

Nodes are seeded automatically in one of the following ways:

- When an unseeded node communicates with a seeded node
- When all the nodes in the cluster are unseeded but able to communicate with each other

VCS requires that you declare the number of nodes that will participate in the cluster. 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, VCS returns the following error:

```
VCS:11037:Node has not received cluster membership yet, cannot
process HA command
```

The number of nodes participating in a cluster could change. A possible scenario is that one or more nodes are down for maintenance when the cluster comes up. In this scenario, the cluster does not seed automatically and therefore VCS does not start successfully. To overcome this issue, you can manually seed the cluster with the currently available nodes.

Before manually seeding the cluster, make sure of the following:

- The nodes participating in the cluster are able to send and receive heartbeats to each other.
- There is no possibility of a network partition condition in the cluster.

Warning: Veritas recommends that you do not seed the cluster manually, unless you are aware of the associated risks and implications.

To manually seed a cluster

- ◆ Run the following command from only one of the operational nodes:

```
gabconfig -x
```

This command seeds all the nodes that communicate with the node on which you run this command.

Note: This seeding is not persistent. If you restart the operational cluster nodes, you will need to rerun this command.

You can also seed a cluster by updating the `%VCS_ROOT%\comms\gab\gabtab.txt` file. By default, VCS records the total number of nodes in the cluster in this file. If the number of nodes that actually participate in the cluster is lower, modify `gabtab.txt` manually.

To manually seed a cluster and make the changes persistent

- 1 Determine the number of nodes that are operational in the cluster.
- 2 For each cluster node, modify the following line in `gabtab.txt`:

```
gabconfig -c -n numberOfNodes
```

Set `numberOfNodes` to the number of currently operational nodes.

When you save `gabtab.txt`, these changes are made persistent. However, for this change to take effect, you need to perform the next step.

- 3
 - To seed the cluster without any application down time, do one of the following:
 - Run the following command on any one operational node:

```
gabconfig -x
```

- Restart the VCS communication stack by running the following commands sequentially on each operational node:

```
taskkill /f /im hashadow.exe
taskkill /f /im had.exe
```

Ensure that the `hashadow` and `had` processes are killed. Then, proceed with the following commands:

```
net stop vcscomm
net stop gab
net stop llt
```

Finally, run the following command on any one operational node:

```
hastart -all
```

- If some application down time is acceptable, reboot each operational node individually so that the changes to `gabtab.txt` take effect.

Reconnecting the private network

When a final network connection is lost, the systems on each side of the network partition segregate into mini-clusters.

Reconnecting a private network after a cluster has been segregated causes HAD to stop and restart.

Following are the rules that determine the systems that will be affected:

- On a two-node cluster, the system with the lowest LLT host ID stays running and the higher recycles HAD.
- In a multi-node cluster, the largest running group stays running. The smaller groups recycle HAD.
- On a multi-node cluster splitting into two equal size clusters, the cluster with the lowest node number stays running. The higher group recycles HAD.

Troubleshooting VCS startup

When VCS is started, GAB, LLT, and HAD are started automatically. If they are not, review the corresponding log file. Startup errors for LLT and GAB are stored in the System Event log. Startup errors for HAD are stored in the Application Event log.

To view log files

- 1 From the Control Panel, double-click **Administrative Tools**, then **Event Viewer**.
- 2 Review the System Log to view LLT and GAB errors.
- 3 Review the Application Log to view HAD errors.

Low Latency Transport (LLT)

During installation, an `llttab.txt` configuration file containing minimum directives is created and placed in the following directory on each node in the cluster:

Drive:\Program Files\VERITAS\comms\llt

Each `llttab.txt` file specifies the node's ID, the network interfaces to use, and other directives.

For the most common LLT directives:

See [“Common LLT directives”](#) on page 479.

Note: The directives must always appear as they are listed in the original default `llttab.txt` file.

Common LLT directives

This topic lists the common LLT directives:

<code>link</code>	<p>Attaches LLT to a network interface. At least one link is required, and up to eight are supported. The first argument to <code>link</code> is a user-defined tag shown in the <code>lltstat</code> output to identify the link. It may also be used in <code>llttab.txt</code> to set optional static MAC addresses.</p> <p>The second argument to <code>link</code> is the device name of the network interface. (To obtain network device names, use the <code>objdir\device</code> command provided by the Windows Device Driver Kit.) There should be one <code>link</code> directive for each network interface, and each network interface configured for LLT must be attached to a separate physical network. LLT uses an unregistered Ethernet SAP of 0xCAFE. If the SAP is unacceptable, refer to the <code>llttab.txt</code> online Help for information on how to customize SAP. Note that IP addresses need not be assigned to the network device because LLT does not use IP addresses.</p>
-------------------	---

link-lowpri	Use this directive in place of <code>link</code> for public network interfaces. This directive prevents VCS communication on the public network until the network is the last link, and reduces the rate of heartbeat broadcasts. Note that LLT distributes network traffic evenly across all available network connections and broadcasts heartbeats to monitor each network connection.
set-cluster	Assigns a unique cluster number. Use this directive when more than one cluster is configured on the same physical network connection. Note that LLT uses a default cluster number of zero.
set-node	Assigns the node ID. This number must be unique for each node in the cluster, and must be between 0-31. Note: LLT fails to operate if nodes share the same ID.
start	This directive must always appear last.

Group Membership Atomic Broadcast (GAB)

During installation, a `gabtab.txt` configuration file is automatically created and placed in the following directory on each system in the cluster:

Drive:\Program Files\VERITAS\comms\gab

Verifying LLT, GAB, and cluster operation

Before verifying LLT, GAB, or cluster operation, you must log on to any node in the cluster using an account with administrator privileges.

Verifying LLT

Use the `lltstat` command to verify the links are active for LLT. This command returns information about the LLT links for the node on which it is typed.

In the following example, `lltstat -n` is typed on System 0 and System 1 in a private network.

System 0

```
Drive:\>lltstat -n
LLT node information:
Node           State    Links
* 0 System 0   OPEN    2
  1 System 1   OPEN    2
```

System 1


```
Drive:\>lltstat -n
LLT node information:
Node           State      Links
0 System 0     OPEN       2
*1 System 1     OPEN       2
```

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

If the output of `lltstat -n` does not show each node in the cluster, or does not show two links for each node, type `lltstat -nvv |` to view additional information about LLT.

In the following example, `lltstat -nvv | more` is typed on System 0 in a private network. Note that each node should be OPEN, each link should be UP, and each address should be correct.

```
Drive:\> lltstat -nvv | more
```

Node	State	Link	Status	Address
* 0 HOUWIN201	OPEN	Adapter0	UP	00:03:47:0D:A8:74
		Adapter1	UP	00:03:47:0D:A8:75
1 HOUWIN202	OPEN	Adapter0	UP	00:03:47:0D:A4:46
		Adapter1	UP	00:03:47:0D:A4:47
2	CONNWAIT	Adapter0	DOWN	
		Adapter1	DOWN	
3	CONNWAIT	Adapter0	DOWN	
		Adapter1	DOWN	
4	CONNWAIT	Adapter0	DOWN	
		Adapter1	DOWN	
5	CONNWAIT	Adapter0	DOWN	
		Adapter1	DOWN	
6	CONNWAIT	Adapter0	DOWN	
		Adapter1	DOWN	
7	CONNWAIT	Adapter0	DOWN	
		Adapter1	DOWN	

8	CONNWAIT	Adapter0	DOWN
		Adapter1	DOWN
9	CONNWAIT	Adapter0	DOWN
		Adapter1	DOWN
10	CONNWAIT	Adapter0	DOWN
		Adapter1	DOWN
12	CONNWAIT	Adapter0	DOWN
		Adapter1	DOWN
13	CONNWAIT	Adapter0	DOWN
		Adapter1	DOWN
14	CONNWAIT	Adapter0	DOWN
		Adapter1	DOWN
15	CONNWAIT	Adapter0	DOWN
		Adapter1	DOWN

-- More --

To obtain information only about the configured systems in the cluster, use the `lltstat -nvv configured` command. See the following example:

Drive:\>**lltstat -nvv configured**

Node	State	Link	Status	Address
* 0 HOUWIN201	OPEN	Adapter0	UP	00:03:47:0D:A8:74
		Adapter1	UP	00:03:47:0D:A8:75
1 HOUWIN202	OPEN	Adapter0	UP	00:03:47:0D:A4:46
		Adapter1	UP	00:03:47:0D:A4:47

To obtain information about the ports open for LLT, type `lltstat -p` on any node. In the following example, `lltstat -p` is typed on System 0 in a private network.

Drive:\> **lltstat -p**

LLT port information:

Port	Usage	Cookie
0	gab	0x0

	opens:	0	1	2	3	4	5	6	7	8	9
10	11	12	13	14...							
	connects:	0	1								

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

Setting the checksum option

Do not use these options unless you are asked to do so by a qualified Technical Support representative.

```
lltconfig -K 0 | 1 | 2
```

Checksum setting options	Action
0	Disable checksums
1	<p>Calculate checksum value for header.</p> <p>When set to 1, LLT checksums each packet it sends to peer to guard against packet corruption on the wire. LLT also offloads checksum calculation to the hardware if the underlying NIC supports it.</p> <p>In case checksum verification fails on the receiver LLT drops that packet, causing the sender to retransmit it.</p>
2	<p>Checksum calculated for header and message.</p> <p>When set to 2, LLT also checksums the whole data buffer submitted by the client to be verified by the peer before delivering it to peer-client. In case the checksum verification fails on the receiver, LLT panics the machine. This is purposefully done to help in analysis of memory corruption from a crash dump.</p>

You can also set the checksum by adding the following information to the lltconfig file

```
set-checksum 0 | 1 | 2
```

Verifying GAB

To verify GAB operation, type the following command as Administrator on each node:

```
Drive:\> gabconfig -a
```

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

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

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

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

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

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

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

Verifying HAD

To verify HAD operation, type the following command as Administrator on each node:

```
Drive:\> gabconfig -a
```

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

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

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

If HAD is not operating, no port membership information is returned.

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

For instructions on how to seed the cluster:

See [“Seeding of VCS clusters”](#) on page 476.

If HAD is running on only one node, the following port membership information is returned:

```
GAB Port Memberships
=====
Port      a      gen      a36e0003      membership 01
Port      h      gen      fd570002      membership 0
Port      h      gen      fd51002       visible ;1
```

This information indicates HAD is running on node 1, but only GAB is running on node 0. Check the Application Event Log on node 0 for more information.

Verifying the cluster

To verify cluster operation, type the following command as Administrator on any node:

```
Drive:\>hasys -display

#System      Attribute      Value
HOUWIN201    AgentsStopped  0
HOUWIN201    AvailableCapacity  100
HOUWIN201    CPUUsage      0
HOUWIN201    CPUUsageMonitoring  Enable 0 ActionThreshold 0
              ActionTimeLimit 0 Action NONE NotifyThreshold 0 NotifyTimeLimit 0
HOUWIN201    Capacity      100
HOUWIN201    ConfigBlockCount  84
HOUWIN201    ConfigChecksum  18907
HOUWIN201    ConfigDiskState  CURRENT
HOUWIN201    ConfigFile     C:\Program
Files\VERITAS\Cluster Server\conf\config
HOUWIN201    ConfigInfoCnt  0
HOUWIN201    ConfigModDate   Tue Dec 03 15:13:58 2001
HOUWIN201    CurrentLimits
HOUWIN201    DiskHbStatus
HOUWIN201    DynamicLoad     0
HOUWIN201    Frozen          0
HOUWIN201    GUIIPAddr
HOUWIN201    LLTNodeId      0
```

```

HOUWIN201      Limits
HOUWIN201      LinkHbStatus      Adapter0 UP Adapter1 UP
HOUWIN201      LoadTimeCounter    0
HOUWIN201      LoadTimeThreshold  600
HOUWIN201      LoadWarningLevel    80
HOUWIN201      MajorVersion        2
HOUWIN201      MinorVersion        0
HOUWIN201      NodeID              0
HOUWIN201      OnGrpCnt            1
HOUWIN201      ShutdownTimeout     60
HOUWIN201      SourceFile          .\main.cf
HOUWIN201      SysInfo             WINNT:HOUWIN201,5.0,2195,
Service Pack  2, INTEL,1
HOUWIN201      SysName
HOUWIN201
HOUWIN201      SysState            RUNNING
HOUWIN201      SystemLocation
HOUWIN201      SystemOwner
HOUWIN201      TFrozen             0
HOUWIN201      TRSE                0
HOUWIN201      UpDownState         Up
HOUWIN201      UserInt             0
HOUWIN201      UserStr             #
HOUWIN202      AgentsStopped       0
HOUWIN202      AvailableCapacity   100
HOUWIN202      CPUUsage            0
HOUWIN202      CPUUsageMonitoring  Enable 0 ActionThreshold 0
ActionTimeLimit 0 Action NONE NotifyThreshold 0NotifyTimeLimit 0
HOUWIN202      Capacity            100
HOUWIN202      ConfigBlockCount    84
HOUWIN202      ConfigCheckSum      18907
HOUWIN202      ConfigDiskState     CURRENT
HOUWIN202      ConfigFile          C:\Program Files\VERITAS\
Cluster Server\conf\config
HOUWIN202      ConfigInfoCnt        0
HOUWIN202      ConfigModDate        Tue Dec 03 15:15:58 2001
HOUWIN202      CurrentLimits
HOUWIN202      DiskHbStatus
HOUWIN202      DynamicLoad         0
HOUWIN202      Frozen              0
HOUWIN202      GUIIPAaddr
HOUWIN202      LLTNodeIdHOUWIN202  Limits
HOUWIN202      LinkHbStatus        Adapter0 UP Adapter1 UP

```

HOUWIN202	LoadTimeCounter	0
HOUWIN202	LoadTimeThreshold	600
HOUWIN202	LoadWarningLevel	80
HOUWIN202	MajorVersion	2
HOUWIN202	MinorVersion	0
HOUWIN202	NodeID	1
HOUWIN202	OnGrpCnt	1
HOUWIN202	ShutdownTimeout	60
HOUWIN202	SourceFile	.\main.cf
HOUWIN202	SysInfo	WINNT:HOUWIN202,5.0,2195,
Service Pack 2,	INTEL,1	
HOUWIN202	SysName	
HOUWIN202		
HOUWIN202	SysState	RUNNING
HOUWIN202	SystemLocation	
HOUWIN202	SystemOwner	
HOUWIN202	TFrozen	0
HOUWIN202	TRSE	0
HOUWIN202	UpDownState	Up
HOUWIN202	UserInt	0
HOUWIN202	UserStr	

Note the value for the attribute ConfigFile is an empty file created by default to enable VCS to start. Also note the value of the attribute SysState is RUNNING, which indicates VCS is started. This output indicates VCS was successfully installed on both nodes in the cluster.

VCS startup errors

This topic includes error messages associated with starting VCS 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 %VCS_HOME%\conf\config.

Type the following commands to verify the configuration and force VCS to reread the configuration file:

```
C:\> cd %VCS_HOME%\conf\config
C:\> hacf -verify .
C:\> 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 %VCS_ROOT%\comms\gab\gabtab.txt. 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 %VCS_ROOT%\comms\llt\llttab.txt. 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.

```
GAB: Port a registration waiting for seed port membership
GAB: Port h registration waiting for seed port membership
```

The following message will also be sent to the engine log:

```
Did not receive cluster membership, manual intervention may be
needed for seeding
```

Seeding the cluster

Perform the following steps to seed the cluster:

To seed the cluster

- 1 Verify the value of `gabconfig -c` in the file %VCS_ROOT%\comms\gab\gabtab.txt is the same for all nodes.
- 2 Determine how many nodes are operational.

- 3 For each cluster node, modify `gabtab.txt` to reflect the required number of members to seed are equal to the number of cluster nodes in operation.
- 4 Reboot each node, or stop HAD -force on all nodes and restart.

See [“Seeding of VCS clusters”](#) on page 476.

Cluster name contains Unicode characters

If Unicode characters are used to define a cluster name, the Veritas High Availability Configuration wizard successfully configures the VCS cluster. However, after the configuration workflow is complete, the VCS cluster transitions to the STALE_ADMIN_WAIT state and fails to start.

This issue occurs because Unicode characters are not permitted in the `main.cf` file. If you encounter this issue, you must change the cluster name to an acceptable value.

To change cluster name

- 1 Take the service group offline.
- 2 Stop the HAD service on all nodes in the cluster using the command:

```
hastop -all -force
```

- 3 Navigate to the `%vcs_home%\conf\config` folder and open the `main.cf` file.
- 4 Change the cluster name and save the `main.cf` file.
- 5 Navigate to `HKEY_LOCAL_MACHINE\SOFTWARE\VERITAS\VCS\Base` in the registry.
- 6 Change the value of `ClusterName (REG_SZ)` to the new cluster name as mentioned in the `main.cf` file.
- 7 On all nodes in the cluster, update the registry key as described in the previous two steps.
- 8 From any node in the cluster, start the HAD service using the command:

```
hastart -all
```

Cluster ID is not unique over a network

Each cluster in a network must have a unique ID. If one or more clusters in a network have the same ID, the VCS cluster fails to start.

If you encounter this issue, you must change the cluster ID to a unique value.

To change cluster ID

- 1 Take the service group offline.
- 2 Stop the HAD service on all nodes in the cluster using the command:

```
hastop -local
```
- 3 Stop the GAB service using the command:

```
net stop gab
```
- 4 Stop the LLT service using the command:

```
net stop llt
```
- 5 Navigate to the `%vcs_root%\comms\llt` folder and open the `llttab.txt` file.
- 6 Update the line `set-cluster old_id` to `set-cluster new_id`, and save and close the `llttab.txt` file.
- 7 Navigate to `HKEY_LOCAL_MACHINE\SOFTWARE\VERITAS\VCS\Base` in the registry.
- 8 Change the value of `ClusterID` (REG_DWORD) to the new cluster ID.
- 9 On all nodes in the cluster, update the registry key as described in the previous two steps.
- 10 From any node in the cluster, start the HAD service using the command:

```
hastart -all
```

Troubleshooting secure clusters

"Error returned from engine: HAD on this node not accepting clients."

This error occurs when an HA command fails because the VCS engine could not initialize its security credentials. When this occurs, the following message is logged to the event log:

"Security ON. Init failed. Clients will be rejected."

Recommended action:

- Make sure that the cluster was configured to run in secure mode before the SecureClus attribute was set to 1.
- Verify that the VCS Authentication Service is running. Stop and restart the service.
- Restart the Veritas High Availability Engine service on the node.

"Unable to connect to the VCS engine securely."

Recommended action:

- Verify that the VCS Authentication Service is running. Stop and restart the service.
- Restart the Veritas High Availability Engine service on the node.

Troubleshooting service groups

This topic cites the most common problems associated with bringing service groups online and taking them offline. 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.

For more information on system states:

See [“System states”](#) on page 521.

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 unfreeze 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.

Warning: 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:

```
C:\> 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 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.

ClusterService group configuration

If you run the `hastop -local` command on a node that is not defined in the ClusterService group's SystemList and has other service groups online, VCS takes the service groups offline on the node and the node gets stuck in the LEAVING state.

To prevent this from happening, make sure that the ClusterService group is defined on all nodes in the cluster. In other words, the SystemList attribute of the ClusterService group must contain all nodes in the cluster.

Troubleshooting resources

This section cites the most common problems associated with bringing resources online and taking them offline. 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: For more information on states:

See [“Remote cluster states”](#) on page 519.

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 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:

```
C:\> 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:

```
C:\> haagent -stop resource_type -sys system
```

To restart the agent, type:

```
C:\> 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.

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.

Troubleshooting and recovery for global clusters

This topic describes the concept of disaster declaration and provides troubleshooting tips for configurations using global clusters.

Disaster declaration

When a cluster in a global cluster transitions to the FAULTED state because it can no longer be contacted, failover executions depend on whether the cause was due to a split-brain, temporary outage, or a permanent disaster at the remote cluster.

If you choose to take action on the failure of a cluster in a global cluster, VCS prompts you to declare the type of failure.

- *Disaster*, implying permanent loss of the primary data center
- *Outage*, implying the primary may return to its current form in some time
- *Disconnect*, implying a split-brain condition; both clusters are up, but the link between them is broken
- *Replica*, implying that data on the takeover target has been made consistent from a backup source and that the RVGPrimary can initiate a takeover when the service group is brought online. This option applies to VVR environments only.

You can select the groups to be failed over to the local cluster, in which case VCS brings the selected groups online on a node based on the group's FailOverPolicy attribute. It also marks the groups as being offline in the other cluster. If you do not

select any service groups to fail over, VCS takes no action except implicitly marking the service groups as offline on the downed cluster.

Lost heartbeats and the inquiry mechanism

The loss of internal and all external heartbeats between any two clusters indicates that the remote cluster is faulted, or that all communication links between the two clusters are broken (a wide-area split-brain).

A configuration with more than two clusters must distinguish between the two (Systems A and B) by querying the remaining clusters to confirm the remote cluster to which heartbeats have been lost is truly down. This mechanism is referred to as "Inquiry." If in a two-cluster configuration a connector loses all heartbeats (internal and external) to the other connector, it must consider the remote cluster faulted. If there are more than two clusters and a connector loses all heartbeats to a second cluster, it queries the remaining connectors regarding their "view" of the cluster in question before declaring it faulted. If the other connectors view the cluster as running (a negative inquiry), the querying connector transitions the cluster to the UNKNOWN state, a process that minimizes false cluster faults. If all connectors report that the cluster is faulted (a positive inquiry), the querying connector also considers it faulted and transitions the remote cluster state to FAULTED.

VCS alerts

VCS alerts are identified by the alert ID, which is comprised of the following elements:

- `alert_type`—The type of the alert.
See [“Types of alerts”](#) on page 497.
- `cluster`—The cluster on which the alert was generated
- `system`—The system on which this alert was generated
- `object`—The name of the VCS object for which this alert was generated. This could be a cluster or a service group.

Alerts are generated in the following format:

```
alert_type-cluster-system-object
```

For example:

```
GNOFAILA-Cluster1-oracle_grp
```

This is an alert of type GNOFAILA generated on cluster Cluster1 for the service group oracle_grp.

Types of alerts

VCS generates the following types of alerts.

- CFAULT—Indicates that a cluster has faulted
- GNOFAILA—Indicates that a global group is unable to fail over within the cluster where it was online. This alert is displayed if the ClusterFailOverPolicy attribute is set to Manual and the wide-area connector (wac) is properly configured and running at the time of the fault.
- GNOFAIL—Indicates that a global group is unable to fail over to any system within the cluster or in a remote cluster.

Some reasons why a global group may not be able to fail over to a remote cluster:

- The ClusterFailOverPolicy is set to either Auto or Connected and VCS is unable to determine a valid remote cluster to which to automatically the group over.
- The ClusterFailOverPolicy attribute is set to Connected and the cluster in which the group has faulted cannot communicate with one or more remote clusters in the group's ClusterList.
- The wide-area connector (wac) is not online or is incorrectly configured in the cluster in which the group has faulted

Managing alerts

Alerts require user intervention. You can respond to an alert in the following ways:

- If the reason for the alert can be ignored, use the Alerts dialog box in the Java or Web consoles or the `haalert` command to delete the alert. You must provide a comment as to why you are deleting the alert; VCS logs the comment to engine log.
- Take an action on administrative alerts that have actions associated with them. You can do so using the Java or Web consoles.
 See [“Actions associated with alerts”](#) on page 498.
- VCS deletes or negates some alerts when a negating event for the alert occurs.
 See [“Negating events”](#) on page 498.

An administrative alert will continue to live if none of the above actions are performed and the VCS engine (HAD) is running on at least one node in the cluster. If HAD is not running on any node in the cluster, the administrative alert is lost.

Actions associated with alerts

This section describes the actions you can perform from the Java and the Web consoles on the following types of alerts:

- CFAULT—When the alert is presented, clicking **Take Action** guides you through the process of failing over the global groups that were online in the cluster before the cluster faulted.
- GNOFAILA—When the alert is presented, clicking **Take Action** guides you through the process of failing over the global group to a remote cluster on which the group is configured to run.
- GNOFAIL—There are no associated actions provided by the consoles for this alert

Negating events

VCS deletes a CFAULT alert when the faulted cluster goes back to the running state

VCS deletes the GNOFAILA and GNOFAIL alerts in response to the following events:

- The faulted group's state changes from FAULTED to ONLINE.
- The group's fault is cleared.
- The group is deleted from the cluster where alert was generated.

Troubleshooting the steward process

When you start the steward, it blocks the command prompt and prints messages to the standard output. To stop the steward, run the following command from a different command prompt of the same system:

If the steward is running in secure mode: `steward -stop - secure`

If the steward is not running in secure mode: `steward -stop`

In addition to the standard output, the steward can log to its own log files:

- `steward_A.log`
- `steward-err_A.log`

Use the `tststew` utility to verify that:

- The steward process is running
- The steward process is sending the right response

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 these utilities.

The getcomms utility

The getcomms utility collects and saves information related to the private network. The information can be used by Technical Support to debug network-related issues.

To run the getcomms utility

- ◆ Run getcomms using the Perl executables provided with VCS.

```
C:\> "VRTS_HOME\VRTSPerl\bin\perl.exe" getcomms.pl [-option]
```

The variable `VRTS_HOME` represents the Veritas installation directory, typically `C:\Program Files\VERITAS`. If you chose the default installation paths, use the following command to run getcomms:

```
C:\> "C:\Program Files\VERITAS\VRTSPerl\bin\perl.exe" getcomms.pl [-option]
```

getcomms options

You have several options to use to limit the diagnostic information to specific components.

[Table 18-2](#) shows the possible options for getcomms.

Table 18-2 getcomms options

Options	Action
-local	Retrieves and dumps information about the local system
-remote	Retrieves and dumps information about all live systems in the cluster
-stuck	Prints the message queue
-d <i>logdir</i>	Dumps information at the directory specified by the variable <i>logdir</i>

Log location

The getcomms utility writes the output to the directory `%temp%\commslog.timestamp` where `%temp%` is a system-defined environment variable and `timestamp` represents the time the log was taken.

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:

```
C:\> hagetcf [-option]
```

Running hagetcf displays output similar to the example below:

```
Veritas Cluster Server(TM) 6.0 for Windows 2012 Diagnostics
Copyright (C)
Dumping output to:
C:\Program Files\Veritas\Cluster Server\hagetcf\<month><day>_<hh><mm>
```

Log location

By default, hagetcf writes the output to the directory `%VCS_HOME%\hagetcf`, where `%VCS_HOME%` is the VCS installation directory, typically `C:\Program Files\VERITAS\Cluster Server\.`

Options for the hagetcf utility

There are several options available with the hagetcf command to limit the diagnostic information to specific components.

Table 18-3 shows the possible options.

Table 18-3 Options for the hagetcf command

Options	Action
-default	Dumps the default VCS logs that include outputs of the following hagetcf command options: -app, -sys, -hw, -ha, -log, -lock, -conf, -state, -islog, -trigger Note: The output also includes information about MSDTC and the VCS agent for MSDTC.

Table 18-3 Options for the hagetcf command (*continued*)

Options	Action
-app	Dumps the application event log.
-sec	Dumps the security event log.
-sys	Dumps the system event log.
-hw	Dumps the hardware event log.
-allevt	Dumps all event logs.
-conf	Dumps the VCS config directory.
-log	Dumps the VCS log directory.
-ldf	Dumps the VCS ldf directory.
-lock	Dumps the lock directory.
-triggers	Dump all files from the VCS triggers directory.
-alldir	Dumps the config, log, ldf, and lock directories.
-ha	Dumps the output of the following commands: <pre> hares -display -all hagrp -display -all hasys -display getcomms.pl </pre>

Table 18-3 Options for the hagetcf command (*continued*)

Options	Action
-state	Dumps the following system state information: <ul style="list-style-type: none"> ■ Dr. Watson log ■ Drive signature information from the havol utility ■ Network information, including NICs, ipconfig, and network-related registry entries ■ The VERITAS registry key ■ Output of the nbstat and the netstat commands ■ Licensing information ■ Disk and volume information ■ SCSI and Fibre configuration information ■ Server configuration information ■ Service and device state and configuration information ■ Processes running on the system ■ Information about the privileges of the current user ■ Information about products installed on the system
-haver	Dumps version information about all VCS binaries.
-nogetcomms	Excludes the output of the getcomms.pl command.
-sql	Dumps information about SQL Server and the VCS agent for SQL Server.
-exch	Dumps information about Exchange Server and the VCS agent for Exchange Server.
-iis	Dumps IIS information.
-notes	Dumps Notes information.
-orac	Dumps information about Oracle and the VCS agent for Oracle.
-msmq	Dumps information about MSMQ.
-allagents	Dumps information about all enterprise agents.
-vxlog	Dumps diagnostic information about the following: <ul style="list-style-type: none"> ■ Backup Exec ■ Veritas Enterprise Administrator
-islog	Dumps installation log.
-o <outdir>	Dumps hagetcf output to <outdir>.

Table 18-3 Options for the `hagetcf` command (*continued*)

Options	Action
<code>-?</code> or <code>-help</code>	Displays the command's usage information.

Note: If you do not specify any options, the command retrieves diagnostic information with the following options: `-app`, `-sys`, `-ha`, `-log`, `-lock`, `-conf`, `-state`, `-islog`, `-trigger`. It also includes the Veritas High Availability Console logs, if the Console components are installed on the cluster system.

The NICTest utility

The NICTest utility determines whether a NIC maintains its connection status in a system-defined variable. The utility helps configuring NICs under VCS.

To perform the NIC test

- 1 At the command prompt, type:

```
C:\> NICTest adapter_macaddress
```

The variable `adapter_macaddress` represents the physical address of the adapter. You can retrieve the MAC address using the `ipconfig -all` command. The utility displays an error message if the entered MAC address is invalid or if it cannot find the specified adapter.

- 2 Press Return.
- 3 Disconnect the NIC and press Return. The system prompts you to connect the NIC.
- 4 Connect the NIC and press Return.

If the NIC does not maintain its connection status, the following message appears:

```
NIC <adapter_macaddress> does not maintain the connection
status.
```

If the NIC maintains its connection status, the following message appears:

```
NIC <adapter_macaddress> maintains the connection status.
Please set the UseConnectionStatus attribute for this
resource to True.
```

The VCSRegUtil utility

If an application is run outside of VCS, registry changes are not logged to the shared disk. VCS provides a utility, VCSRegUtil.exe, that marks the system on which registry changes are made outside of the VCS environment.

If a system is marked by the VCSRegUtil utility, the agent detects registry changes when VCS is started. The agent then logs changes to the shared disk. Therefore, you must run the VCSRegUtil.exe utility whenever you run the clustered application outside of VCS. For example, you must use it when issuing the command `hastop -all -force` to take all resources offline and run the application outside the VCS environment. The utility also unmarks a previously marked system. When the resource is brought online on a system marked by this utility, the agent unmarks the system.

Note: If a system is marked using VCSRegUtil.exe, and if the attribute `RestoreLocally` is set to 1, the system marking overrides the `RestoreLocally` attribute and registry changes are not applied to the system when it is brought online.

To mark a system, at the command prompt, type:

```
C:\> VCSRegUtil /RESOURCE=RegRepResourceName /MARK
```

To unmark a system, at the command prompt, type:

```
C:\> VCSRegUtil /RESOURCE=RegRepResourceName /UNMARK
```

The havol utility

The havol utility provides the following options:

- `-getdrive`: Retrieves information about the disk and stores it in a file called `DriveInfo.txt` in the same path from where you executed the command.
- `-scsitest`: Reserves and releases disks. It helps troubleshoot issues related to SCSI reservation.

Note: `-scsitest` option is not supported in an SFW environment.

```
C:\> havol -scsitest <options>
C:\> havol -getdrive [-details]
```

Using the `-getdrive` option

At the command prompt, type:


```
C:\> havol -getdrive
```

For detailed information about the disk, type:

```
C:\> havol -getdrive -details
```

The utility retrieves information about the disk and stores it in a file called DriveInfo.txt in the same path from where you executed the command.

Sample file contents include:

Detailed Information about Fixed Disks with valid volumes
in the system: VCSW2K112J

```

Harddisk Number = 1
Harddisk Type = Basic Disk
Disk Signature = 130349684
Valid Partitions = 3
Access Test = SUCCESS
Partition Number = 3
Partition Type = IFS
Partition Bootable = NO
Partition Size = 400.06 MB
WINNT style Name = \device\Harddisk1\Partition3
Target Name = \Device\HarddiskVolume6
Device Name =
\\?\Volume{03074b0e-b4d7-11d6-b5a9-00d0b7471a1f}\
DriveLetter = Unassigned
DrivePath001 = F:\f1\

Partition Number = 2
Partition Type = IFS Partition
Bootable = NO
Partition Size = 400.06 MB
WINNT style Name = \device\Harddisk1\Partition2
Target Name = \Device\HarddiskVolume5
Device Name =
\\?\Volume{03074af7-b4d7-11d6-b5a9-00d0b7471a1f}\
DriveLetter = Unassigned
DrivePath001 = F:\f2\

Partition Number = 1
Partition Type = IFS
Partition Bootable = NO
Partition Size = 4.01 GB

```

```

WINNT style Name = \device\Harddisk1\Partition1
Target Name = \Device\HarddiskVolume4
Device Name =
\\?\Volume{e587ddc7-8cee-11d6-b401-00d0b7471a1f}\
DriveLetter = F:
MountPoint001 = F:\f2\ ->
\\?\Volume{03074af7-b4d7-11d6-b5a9-00d0b7471a1f}\
MountPoint002 = F:\f1\ ->
\\?\Volume{03074b0e-b4d7-11d6-b5a9-00d0b7471a1f}\

```

Using the -scsittest option

At the command prompt, type:

```
C:\> havol -scsittest [/option]
```

The variable *option* represents additional parameters for the command.

See [“The -scsittest command options”](#) on page 506.

Warning: Reserving or releasing shared disks may cause the configured service groups to fail over to another system.

Note: -scsittest option is not supported in an SFW environment.

Retrieving the disk number

Most scsittest options require the disk number. To list the disk numbers visible from the system, type the following command:

```
C:\> havol -scsittest /L
```

Verify the disk signature to ensure you choose the correct disk number. If the required disk number or signature is not listed, try rescanning the SCSI bus. Type:

```
C:\> havol -scsittest /RESCAN
```

The -scsittest command options

You have several options with the scsittest command to limit the diagnostic information to specific components.

[Table 18-4](#) shows the basic options for the -scsittest command.

Table 18-4 -scsitest command: basic options

Basic Options	Action
-ADDR:1	Gets the SCSI address of disk number 1.
-LISTDISKS or -L	Lists all visible disks.
-REL:1	Releases disk number 1.
-RES:1	Reserves disk number 1.
-RESCAN	Rescans the SCSI bus.
-RESET:1	Resets the disk number 1 (in ioctl mode).
-SIG:1	Retrieves the signature of disk number 1.

Table 18-5 shows the advanced options for the `-scscitest` command.

Table 18-5 -scsitest command: advanced options

Advanced Options	Action
-DISKCOUNT	Returns the total number of disks reserved persistently.
-PREL:1	Persistently releases disk number 1.
-PRES:1	Persistently reserves disk number 1.
-REMOVEALL	Removes all disks from persistent reservation.
-RESETPBI:1,0	Resets the port number 1 and path 0 by ioctl mode.
-RESETPBS:1,0	Resets the port number 1 and path 0 by SRB mode.
-STARTDRV	Starts the DiskRes driver.
-STOPDRV	Stops the DiskRes driver.
-VER	Retrieves the DiskRes.sys version.

The vmgetdrive utility

Use the VMGetDrive utility to retrieve information about the cluster disk groups and configured volumes.

Note: You can run this utility successfully only if SFW is installed, failing which an error of `VxCMD.DLL` file is missing is displayed.

To retrieve information about the cluster disk groups using the `VMGetDrive` utility

- 1 At the command prompt, from the path `%VCS_HOME%\bin`, type:

```
%VCS_HOME%\bin> vmgetdrive
```

The system retrieves information about the volume and stores it in a file called `VMDriveInfo.txt` in the same path from where you executed the command.

- 2 Open the file `VMDriveInfo.txt` using a text editor, and get the required information. Sample file contents include:

```
There are 1 Imported Cluster Disk Groups
```

```
DiskGroup Name = VCS1
```

```
No of disks in diskgroup 'VCS1' = 2
```

```
    Harddisk2
```

```
    Harddisk3
```

```
No of volumes in diskgroup 'VCS1' = 2
```

```
Volume Name = Stripel
```

```
    Drive Letter = NONE
```

```
    File System = NTFS
```

```
    Mount Point = NONE
```

```
    Volume Label =
```

```
    Volume Type = STRIPED
```

```
Volume Name = Volumel
```

```
    Drive Letter = NONE
```

```
    File System = NTFS
```

```
    Mount Point = NONE
```

```
    Volume Label =
```

```
Volume Type = CONCATENATED
```

Configuring the VCS Helper service manually

Use the `HAadHelper` command to configure the VCS Helper service manually.

Command syntax

Following is the command syntax for the `HAadHelper` command:

```
HADHelper /Install /User:<user_name> [/Password:<password>]
HADHelper /Uninstall
HADHelper /Configure /User:<user_name> [/Password:<password>]
HADHelper /ShowConfig
```

- If you do not specify a password for the `/Install` and `/Configure` options, the command prompts for a password.
- Specify the `<user_name>` as `domain\user` or `user`. If you do not append the domain name, the command assumes the user belongs to the current domain.
- Assign the privilege **Add workstation to domain** on the domain controller to the user.

Command options

Note that the following command options are case insensitive.

[Table 18-6](#) shows the possible options for the `HadHelper` command.

Table 18-6 HadHelper command options

Options	Action
<code>/Install</code>	Installs the VCS Helper service (HADHelper), configures the user for the service, assigns the required local security privileges to the user, and adds the user to the local administrators group. If the service already exists, the option configures the user for the service.
<code>/Uninstall</code>	Uninstalls the service, removes the local security privileges for the configured user, and removes the user from local administrators group. Note: Stop the service before running the command to uninstall the service.
<code>/Configure</code>	Changes the user for the service, assigns the required local security privileges to the user, and adds the user to local administrators group. The option also revokes the local security privileges of the previous user and removes the user from local administrators group.
<code>/ShowConfig</code>	Displays the user name, user SID, and the local security privileges held by the user.

Appendixes

- [Appendix A. VCS user privileges—administration matrices](#)
- [Appendix B. Cluster and system states](#)
- [Appendix C. VCS attributes](#)
- [Appendix D. Configuring LLT over UDP](#)
- [Appendix E. Handling concurrency violation in any-to-any configurations](#)
- [Appendix F. Accessibility and VCS](#)

VCS user privileges—administration matrices

This appendix includes the following topics:

- [About administration matrices](#)
- [Administration matrices](#)

About administration matrices

In general, users with Guest privileges can run the following command options: -display, -state, and -value.

Users with privileges for Group Operator and Cluster Operator can execute the following options: -online, -offline, and -switch.

Users with Group Administrator and Cluster Administrator privileges can run the following options -add, -delete, and -modify.

See [“About VCS user privileges and roles”](#) on page 66.

Administration matrices

Review the matrices in the following topics to determine which command options can be executed within a specific user role. Checkmarks denote the command and option can be executed. A dash indicates they cannot.

Agent Operations (haagent)

[Table A-1](#) lists agent operations and required privileges.

Table A-1 User privileges for agent operations

Agent Operation	Guest	Group Operator	Group Admin.	Cluster Operator	Cluster Admin.
Start agent	–	–	–	✓	✓
Stop agent	–	–	–	✓	✓
Display info	✓	✓	✓	✓	✓
List agents	✓	✓	✓	✓	✓

Attribute Operations (haattr)

[Table A-2](#) lists attribute operations and required privileges.

Table A-2 User privileges for attribute operations

Attribute Operations	Guest	Group Operator	Group Admin.	Cluster Operator	Cluster Admin.
Add	–	–	–	–	✓
Change default value	–	–	–	–	✓
Delete	–	–	–	–	✓
Display	✓	✓	✓	✓	✓

Cluster Operations (haclus, haconf)

[Table A-3](#) lists cluster operations and required privileges.

Table A-3 User privileges for cluster operations

Cluster Operations	Cluster Guest	Group Operator	Group Admin.	Cluster Operator	Cluster Admin.
Display	✓	✓	✓	✓	✓
Modify	–	–	–	–	✓
Add	–	–	–	–	✓

Table A-3 User privileges for cluster operations (*continued*)

Cluster Operations	Cluster Guest	Group Operator	Group Admin.	Cluster Operator	Cluster Admin.
Delete	—	—	—	—	✓
Declare	—	—	—	✓	✓
View state or status	✓	✓	✓	✓	✓
Update license	—	—	—	—	✓
Make configuration read-write	—	—	✓	—	✓
Save configuration	—	—	✓	—	✓
Make configuration read-only	—	—	✓	—	✓

Service group operations (hagrp)

[Table A-4](#) lists service group operations and required privileges.

Table A-4 User privileges for service group operations

Service Group Operations	Guest	Group Operator	Group Admin.	Cluster Operator	Cluster Admin.
Add and delete	—	—	—	—	✓
Link and unlink	—	—	—	—	✓
Clear	—	✓	✓	✓	✓
Bring online	—	✓	✓	✓	✓
Take offline	—	✓	✓	✓	✓
View state	✓	✓	✓	✓	✓
Switch	—	✓	✓	✓	✓
Freeze/unfreeze	—	✓	✓	✓	✓
Freeze/unfreeze persistent	—	—	✓	—	✓
Enable	—	—	✓	—	✓

Table A-4 User privileges for service group operations (*continued*)

Service Group Operations	Guest	Group Operator	Group Admin.	Cluster Operator	Cluster Admin.
Disable	–	–	✓	–	✓
Modify	–	–	✓	–	✓
Display	✓	✓	✓	✓	✓
View dependencies	✓	✓	✓	✓	✓
View resources	✓	✓	✓	✓	✓
List	✓	✓	✓	✓	✓
Enable resources	–	–	✓	–	✓
Disable resources	–	–	✓	–	✓
Flush	–	✓	✓	✓	✓
Autoenable	–	✓	✓	✓	✓
Ignore	–	✓	✓	✓	✓

Heartbeat operations (hahb)

[Table A-5](#) lists heartbeat operations and required privileges.

Table A-5 User privileges for heartbeat operations

Heartbeat Operations	Guest	Group Operator	Group Admin.	Cluster Operator	Cluster Admin.
Add	–	–	–	–	✓
Delete	–	–	–	–	✓
Make local	–	–	–	–	✓
Make global	–	–	–	–	✓
Display	✓	✓	✓	✓	✓
View state	✓	✓	✓	✓	✓
List	✓	✓	✓	✓	✓

Log operations (halog)

[Table A-6](#) lists log operations and required privileges.

Table A-6 User privileges for log operations

Log operations	Guest	Group Operator	Group Admin.	Cluster Operator	Cluster Admin.
Enable debug tags	–	–	–	–	✓
Delete debug tags	–	–	–	–	✓
Add messages to log file	–	–	✓	–	✓
Display	✓	✓	✓	✓	✓
Display log file info	✓	✓	✓	✓	✓

Resource operations (hares)

[Table A-7](#) lists resource operations and required privileges.

Table A-7 User privileges for resource operations

Resource operations	Guest	Group Operator	Group Admin.	Cluster Operator	Cluster Admin.
Add	–	–	✓	–	✓
Delete	–	–	✓	–	✓
Make attribute local	–	–	✓	–	✓
Make attribute global	–	–	✓	–	✓
Link and unlink	–	–	✓	–	✓
Clear	–	✓	✓	✓	✓
Bring online	–	✓	✓	✓	✓
Take offline	–	✓	✓	✓	✓
Modify	–	–	✓	–	✓
View state	✓	✓	✓	✓	✓

Table A-7 User privileges for resource operations (*continued*)

Resource operations	Guest	Group Operator	Group Admin.	Cluster Operator	Cluster Admin.
Display	✓	✓	✓	✓	✓
View dependencies	✓	✓	✓	✓	✓
List, Value	✓	✓	✓	✓	✓
Probe	–	✓	✓	✓	✓
Override attribute	–	–	✓	–	✓
Remove overrides	–	–	✓	–	✓
Run an action	–	✓	✓	✓	✓
Refresh info	–	✓	✓	✓	✓
Flush info	–	✓	✓	✓	✓

System operations (hasys)

[Table A-8](#) lists system operations and required privileges.

Table A-8 User privileges for system operations

System operations	Guest	Group Operator	Group Admin.	Cluster Operator	Cluster Admin.
Add	–	–	–	–	✓
Delete	–	–	–	–	✓
Freeze and unfreeze	–	–	–	✓	✓
Freeze and unfreeze persistent	–	–	–	–	✓
Freeze and evacuate	–	–	–	–	✓
Display	✓	✓	✓	✓	✓
Start forcibly	–	–	–	–	✓
Modify	–	–	–	–	✓

Table A-8 User privileges for system operations (*continued*)

System operations	Guest	Group Operator	Group Admin.	Cluster Operator	Cluster Admin.
View state	✓	✓	✓	✓	✓
List	✓	✓	✓	✓	✓
Update license	—	—	—	—	✓

Resource type operations (hatype)

[Table A-9](#) lists resource type operations and required privileges.

Table A-9 User privileges for resource type operations

Resource type operations	Guest	Group Operator	Group Admin.	Cluster Operator	Cluster Admin.
Add	—	—	—	—	✓
Delete	—	—	—	—	✓
Display	✓	✓	✓	✓	✓
View resources	✓	✓	✓	✓	✓
Modify	—	—	—	—	✓
List	✓	✓	✓	✓	✓

User operations (hauser)

[Table A-10](#) lists user operations and required privileges.

Table A-10 User privileges for user operations

User operations	Guest	Group Operator	Group Admin.	Cluster Operator	Cluster Admin.
Add	—	—	—	—	✓
Delete	—	—	—	—	✓

Table A-10 User privileges for user operations *(continued)*

User operations	Guest	Group Operator	Group Admin.	Cluster Operator	Cluster Admin.
Update	✓ Note: If configuration is read/write	✓ Note: If configuration is read/write	✓	✓ Note: If configuration is read/write	✓
Display	✓	✓	✓	✓	✓
List	✓	✓	✓	✓	✓
Modify privileges	—	—	✓	—	✓

Cluster and system states

This appendix includes the following topics:

- [Remote cluster states](#)
- [System states](#)

Remote cluster states

In global clusters, the "health" of the remote clusters is monitored and maintained by the wide-area connector process. The connector process uses heartbeats, such as `lcmp`, to monitor the state of remote clusters. The state is then communicated to HAD, which then uses the information to take appropriate action when required. For example, when a cluster is shut down gracefully, the connector transitions its local cluster state to `EXITING` and notifies the remote clusters of the new state. When the cluster exits and the remote connectors lose their TCP/IP connection to it, each remote connector transitions their view of the cluster to `EXITED`.

To enable wide-area network heartbeats, the wide-area connector process must be up and running. For wide-area connectors to connect to remote clusters, at least one heartbeat to the specified cluster must report the state as `ALIVE`.

There are three heartbeat states for remote clusters: `HBUNKNOWN`, `HBALIVE`, and `HBDEAD`.

See [“Examples of system state transitions”](#) on page 523.

[Table B-1](#) provides a list of VCS remote cluster states and their descriptions.

Table B-1 VCS state definitions

State	Definition
INIT	The initial state of the cluster. This is the default state.

Table B-1 VCS state definitions (*continued*)

State	Definition
BUILD	The local cluster is receiving the initial snapshot from the remote cluster.
RUNNING	Indicates the remote cluster is running and connected to the local cluster.
LOST_HB	The connector process on the local cluster is not receiving heartbeats from the remote cluster
LOST_CONN	The connector process on the local cluster has lost the TCP/IP connection to the remote cluster.
UNKNOWN	The connector process on the local cluster determines the remote cluster is down, but another remote cluster sends a response indicating otherwise.
FAULTED	The remote cluster is down.
EXITING	The remote cluster is exiting gracefully.
EXITED	The remote cluster exited gracefully.
INQUIRY	The connector process on the local cluster is querying other clusters on which heartbeats were lost.
TRANSITIONING	The connector process on the remote cluster is failing over to another node in the cluster.

Examples of cluster state transitions

Following are examples of cluster state transitions:

- If a remote cluster joins the global cluster configuration, the other clusters in the configuration transition their "view" of the remote cluster to the RUNNING state:
INIT -> BUILD -> RUNNING
- If a cluster loses all heartbeats to a remote cluster in the RUNNING state, inquiries are sent. If all inquiry responses indicate the remote cluster is actually down, the cluster transitions the remote cluster state to FAULTED:
RUNNING -> LOST_HB -> INQUIRY -> FAULTED
- If at least one response does not indicate the cluster is down, the cluster transitions the remote cluster state to UNKNOWN:
RUNNING -> LOST_HB -> INQUIRY -> UNKNOWN
- When the ClusterService service group, which maintains the connector process as highly available, fails over to another system in the cluster, the remote clusters

transition their view of that cluster to **TRANSITIONING**, then back to **RUNNING** after the failover is successful:

RUNNING -> TRANSITIONING -> BUILD -> RUNNING

- When a remote cluster in a **RUNNING** state is stopped (by taking the **ClusterService** service group offline), the remote cluster transitions to **EXITED**:
RUNNING -> EXITING -> EXITED

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 invalid, the engine transitions to the state **LOCAL_BUILD**, and builds the configuration from disk. If the configuration is invalid, the system transitions to the state of **STALE_ADMIN_WAIT**.

See [“Examples of system state transitions”](#) on page 523.

[Table B-2](#) provides a list of VCS system states and their descriptions.

Table B-2 VCS system states

State	Definition
ADMIN_WAIT	The running configuration was lost. A system transitions into this state for the following reasons: <ul style="list-style-type: none">■ The last system in the RUNNING configuration leaves the cluster before another system takes a snapshot of its configuration and transitions to the RUNNING state.■ A system in LOCAL_BUILD state tries to build the configuration from disk and receives an unexpected error from hacf indicating the configuration is invalid.
CURRENT_ DISCOVER_WAIT	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.

Table B-2 VCS system states (*continued*)

State	Definition
CURRENT_PEER_WAIT	The system has a valid 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.
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	<p>The system has an invalid 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.</p> <p>Systems in STALE_ADMIN_WAIT transition to STALE_PEER_WAIT.</p>
STALE_DISCOVER_WAIT	The system has joined the cluster with an invalid 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 an invalid 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 system state transitions

Following are examples of system 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 an invalid 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 an invalid 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.
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 an invalid 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 `hastop` 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

VCS attributes

This appendix includes the following topics:

- [About attributes and their definitions](#)
- [Resource attributes](#)
- [Resource type attributes](#)
- [Service group attributes](#)
- [System attributes](#)
- [Cluster attributes](#)
- [Heartbeat attributes \(for global clusters\)](#)
- [Remote cluster attributes](#)

About attributes and their definitions

You can modify the values of attributes labelled user-defined from the command line or graphical user interface, or by manually modifying the `main.cf` configuration file. You can change the default values to better suit your environment and enhance performance.

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 labelled 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.

Attribute values are case-sensitive.

See [“About VCS attributes”](#) on page 56.

Resource attributes

[Table C-1](#) lists resource attributes.

Table C-1 Resource attributes

Resource attributes	Description
ArgListValues (agent-defined)	<p>List of arguments passed to the resource's agent on each system. This attribute is resource-specific and system-specific, meaning that the list of values passed to the agent depend on which system and resource they are intended.</p> <p>The number of values in the ArgListValues should not exceed 425. This requirement becomes a consideration if an attribute in the ArgList is a keylist, a vector, or an association. Such type of non-scalar attributes can typically take any number of values, and when they appear in the ArgList, the agent has to compute ArgListValues from the value of such attributes. If the non-scalar attribute contains many values, it will increase the size of ArgListValues. Hence when developing an agent, this consideration should be kept in mind when adding a non-scalar attribute in the ArgList. Users of the agent need to be notified that the attribute should not be configured to be so large that it pushes that number of values in the ArgListValues attribute to be more than 425.</p> <ul style="list-style-type: none">■ Type and dimension: string-vector■ Default: non-applicable.

Table C-1 Resource attributes (*continued*)

Resource attributes	Description
AutoStart (user-defined)	<p>Indicates if a resource should be brought online as part of a service group online, or if it needs the <code>hares -online</code> command.</p> <p>For example, you have two resources, R1 and R2. R1 and R2 are in group G1. R1 has an AutoStart value of 0, R2 has an AutoStart value of 1.</p> <p>In this case, you see the following effects:</p> <pre># hagr -online G1 -sys sys1</pre> <p>Brings only R2 to an ONLINE state. The group state is ONLINE and not a PARTIAL state. R1 remains OFFLINE.</p> <pre># hares -online R1 -sys sys1</pre> <p>Brings R1 online, the group state is ONLINE.</p> <pre># hares -offline R2 -sys sys1</pre> <p>Brings R2 offline, the group state is PARTIAL.</p> <p>Resources with a value of zero for AutoStart, contribute to the group's state only in their ONLINE state and not for their OFFLINE state.</p> <ul style="list-style-type: none"> ■ Type and dimension: boolean-scalar ■ Default: 1
ComputeStats (user-defined)	<p>Indicates to agent framework whether or not to calculate the resource's monitor statistics.</p> <ul style="list-style-type: none"> ■ Type and dimension: boolean-scalar ■ Default: 0
ConfidenceLevel (agent-defined)	<p>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.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: 0
Critical (user-defined)	<p>Indicates whether a fault of this resource should trigger a failover of the entire group or not. If Critical is 0 and no parent above has Critical = 1, then the resource fault will not cause group failover.</p> <ul style="list-style-type: none"> ■ Type and dimension: boolean-scalar ■ Default: 1

Table C-1 Resource attributes (*continued*)

Resource attributes	Description
Enabled (user-defined)	<p>Indicates agents monitor the resource.</p> <p>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.</p> <p>When Enabled is set to 0, it implies a disabled resource.</p> <ul style="list-style-type: none"> ■ Type and dimension: boolean-scalar ■ Default: If you specify the resource in <code>main.cf</code> prior to starting VCS, the default value for this attribute is 1, otherwise it is 0.
Flags (system use only)	<p>Provides additional information for the state of a resource. Primarily this attribute raises flags pertaining to the resource. Values:</p> <p>ADMIN WAIT—The running configuration of a system is lost.</p> <p>RESTARTING —The agent is attempting to restart the resource because the resource was detected as offline in latest monitor cycle unexpectedly. See <code>RestartLimit</code> attribute for more information.</p> <p>STATE UNKNOWN—The latest monitor call by the agent could not determine if the resource was online or offline.</p> <p>MONITOR TIMEDOUT —The latest monitor call by the agent was terminated because it exceeded the maximum time specified by the static attribute <code>MonitorTimeout</code>.</p> <p>UNABLE TO OFFLINE—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 clean function completes successfully, but the subsequent monitor hangs or is unable to determine resource status.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: Not applicable.
Group (system use only)	<p>String name of the service group to which the resource belongs.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-scalar ■ Default: Not applicable.

Table C-1 Resource attributes (*continued*)

Resource attributes	Description
IState (system use only)	<p>The internal state of a resource. In addition to the State attribute, this attribute shows to which state the resource is transitioning. 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 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 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 (path) - Agent notified to take the resource offline but procedure not yet complete. When the procedure completes, the resource's children which are a member of the path in the dependency tree will also be offline.</p> <p>WAITING TO GO OFFLINE (reverse) - Resource waiting to be brought offline, but when it is offline it attempts to go online. Typically this is the result of issuing an online command while resource was waiting to go offline.</p> <p>WAITING TO GO ONLINE (reverse/path) - Resource waiting to be brought online, but when online it is brought offline. Resource transitions to WAITING TO GO OFFLINE (path). Typically this is the result of fault of a child resource while resource was waiting to go online.</p> <p>WAITING FOR PARENT OFFLINE – Resource waiting for parent resource to go offline. When parent is offline the resource is brought offline.</p> <p>Note: Although this attribute accepts integer types, the command line indicates the text representations.</p>

Table C-1 Resource attributes (*continued*)

Resource attributes	Description
IState (system use only)	<p>WAITING TO GO ONLINE (reverse/propagate)—Same as above, but resource propagates the offline operation.</p> <p>IStates on the source system for migration operations:</p> <ul style="list-style-type: none"> ■ WAITING FOR OFFLINE VALIDATION (migrate) – This state is applicable for resource on source system and indicates that migration operation has been accepted and VCS is validating whether migration is possible. ■ WAITING FOR MIGRATION OFFLINE – This state is applicable for resource on source system and indicates that migration operation has passed the prerequisite checks and validations on the source system. ■ WAITING TO COMPLETE MIGRATION – This state is applicable for resource on source system and indicates that migration process is complete on the source system and the VCS engine is waiting for the resource to come online on target system. <p>IStates on the target system for migration operations:</p> <ul style="list-style-type: none"> ■ WAITING FOR ONLINE VALIDATION (migrate) – This state is applicable for resource on target system and indicates that migration operations are accepted and VCS is validating whether migration is possible. ■ WAITING FOR MIGRATION ONLINE – This state is applicable for resource on target system and indicates that migration operation has passed the prerequisite checks and validations on the source system. ■ WAITING TO COMPLETE MIGRATION (online) – This state is applicable for resource on target system and indicates that migration process is complete on the source system and the VCS engine is waiting for the resource to come online on target system. ■ Type and dimension: integer-scalar ■ Default: 1 NOT WAITING
LastOnline (system use only)	<p>Indicates the system name on which the resource was last online. This attribute is set by VCS.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-scalar ■ Default: Not applicable

Table C-1 Resource attributes (*continued*)

Resource attributes	Description
ManageFaults (user-defined)	<p>Specifies whether VCS responds to a resource fault by calling the Clean entry point. Its value supersedes all the values assigned to the attribute at service group level.</p> <p>This attribute can take the following values:</p> <ul style="list-style-type: none"> ■ ACT: VCS invokes the Clean function with <code>CleanReason</code> set to Online Hung. ■ IGNORE: VCS changes the resource state to ONLINE ADMIN_WAIT. ■ NULL (Blank): VCS takes action based on the values set for the attribute at the service group level. <p>Default value: ""</p>
MonitorOnly (system use only)	<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 only be monitored.</p> <p>Note: This attribute can only be affected by the command <code>hagrp -freeze</code>.</p> <ul style="list-style-type: none"> ■ Type and dimension: boolean-scalar ■ Default: 0
MonitorTimeStats (system use only)	<p>Valid keys are Average and TS. Average is the average time taken by the monitor function over the last Frequency number of monitor cycles. TS is the timestamp indicating when the engine updated the resource's Average value.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-association ■ Default: Average = 0 TS = ""
Name (system use only)	<p>Contains the actual name of the resource.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-scalar ■ Default: Not applicable.
Path (system use only)	<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> <ul style="list-style-type: none"> ■ Type and dimension: boolean-scalar ■ Default: 0
Probed (system use only)	<p>Indicates whether the state of the resource has been determined by the agent by running the monitor function.</p> <ul style="list-style-type: none"> ■ Type and dimension: boolean-scalar ■ Default: 0

Table C-1 Resource attributes (*continued*)

Resource attributes	Description
ResourceInfo (system use only)	<p>This attribute has three predefined keys: State: values are Valid, Invalid, or Stale. Msg: output of the info agent function of the resource on stdout by the agent framework. TS: timestamp indicating when the ResourceInfo attribute was updated by the agent framework</p> <ul style="list-style-type: none"> ■ Type and dimension: string-association ■ Default: State = Valid Msg = "" TS = ""
ResourceOwner (user-defined)	<p>This attribute is used for VCS email notification and logging. VCS sends email notification to the person that is designated in this attribute when events occur that are related to the resource. Note that while VCS logs most events, not all events trigger notifications. VCS also logs the owner name when certain events occur.</p> <p>Make sure to set the severity level at which you want notifications to be sent to ResourceOwner or to at least one recipient defined in the SmtprRecipients attribute of the NotifierMngr agent.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-scalar ■ Default: "" ■ Example: "jdoe@example.com"
ResourceRecipients (user-defined)	<p>This attribute is used for VCS email notification. VCS sends email notification to persons designated in this attribute when events related to the resource occur and when the event's severity level is equal to or greater than the level specified in the attribute.</p> <p>Make sure to set the severity level at which you want notifications to be sent to ResourceRecipients or to at least one recipient defined in the SmtprRecipients attribute of the NotifierMngr agent.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-association ■ email id: The e-mail address of the person registered as a recipient for notification. severity: The minimum level of severity at which notifications must be sent.
Signaled (system use only)	<p>Indicates whether a resource has been traversed. Used when bringing a service group online or taking it offline.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-association ■ Default: Not applicable.
Start (system use only)	<p>Indicates whether a resource was started (the process of bringing it online was initiated) on a system.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer -scalar ■ Default: 0

Table C-1 Resource attributes (continued)

Resource attributes	Description
State (system use only)	<p>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:</p> <p>ONLINE</p> <p>OFFLINE</p> <p>FAULTED</p> <p>OFFLINE MONITOR TIMEDOUT</p> <p>OFFLINE STATE UNKNOWN</p> <p>OFFLINE ADMIN WAIT</p> <p>ONLINE RESTARTING</p> <p>ONLINE MONITOR TIMEDOUT</p> <p>ONLINE STATE UNKNOWN</p> <p>ONLINE UNABLE TO OFFLINE</p> <p>ONLINE ADMIN WAIT</p> <p>FAULTED MONITOR TIMEDOUT</p> <p>FAULTED STATE UNKNOWN</p> <p>A FAULTED resource is physically offline, though unintentionally.</p> <p>Note: Although this attribute accepts integer types, the command line indicates the text representations.</p> <p>Type and dimension: integer -scalar</p> <p>Default: 0</p>
TriggerEvent (user-defined)	<p>A flag that turns Events on or off.</p> <ul style="list-style-type: none">■ Type and dimension: boolean-scalar■ Default: 0

Table C-1 Resource attributes (*continued*)

Resource attributes	Description
TriggerPath (user-defined)	<p>Enables you to customize the trigger path.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-scalar ■ Default: "" <p>If a trigger is enabled but the trigger path at the service group level and at the resource level is "" (default), VCS invokes the trigger from the \$VCS_HOME/bin/<i>triggers</i> directory.</p> <p>The TriggerPath value is case-sensitive. VCS does not trim the leading spaces or trailing spaces in the Trigger Path value. If the path contains leading spaces or trailing spaces, the trigger might fail to get executed. The path that you specify is relative to \$VCS_HOME and the trigger path defined for the service group.</p> <p>Specify the path in the following format:</p> <p><i>ServiceGroupTriggerPath/Resource/Trigger</i></p> <p>If TriggerPath for service group sg1 is mytriggers/sg1 and TriggerPath for resource res1 is "", you must store the trigger script in the \$VCS_HOME/mytriggers/sg1/res1 directory. For example, store the resstatechange trigger script in the \$VCS_HOME/mytriggers/sg1/res1 directory. You can manage triggers for all resources for a service group more easily.</p> <p>If TriggerPath for resource res1 is mytriggers/sg1/vip1 in the preceding example, you must store the trigger script in the \$VCS_HOME/mytriggers/sg1/vip1 directory. For example, store the resstatechange trigger script in the \$VCS_HOME/mytriggers/sg1/vip1 directory.</p> <p>Modification of TriggerPath value at the resource level does not change the TriggerPath value at the service group level. Likewise, modification of TriggerPath value at the service group level does not change the TriggerPath value at the resource level.</p>
TriggerResRestart (user-defined)	<p>Determines whether or not to invoke the resrestart trigger if resource restarts.</p> <p>See “About the resrestart event trigger” on page 381.</p> <p>If this attribute is enabled at the group level, the resrestart trigger is invoked irrespective of the value of this attribute at the resource level.</p> <p>See “Service group attributes” on page 546.</p> <ul style="list-style-type: none"> ■ Type and dimension: boolean-scalar ■ Default: 0 (disabled)

Table C-1 Resource attributes (*continued*)

Resource attributes	Description
TriggerResStateChange (user-defined)	<p>Determines whether or not to invoke the resstatechange trigger if the resource changes state.</p> <p>See “About the resstatechange event trigger” on page 381.</p> <p>If this attribute is enabled at the group level, then the resstatechange trigger is invoked irrespective of the value of this attribute at the resource level.</p> <p>See “Service group attributes” on page 546.</p> <ul style="list-style-type: none">■ Type and dimension: boolean-scalar■ Default: 0 (disabled)
TriggersEnabled (user-defined)	<p>Determines if a specific trigger is enabled or not.</p> <p>Triggers are disabled by default. You can enable specific triggers on all nodes or only on selected nodes. Valid values are RESFAULT, RESNOTOFF, RESSTATECHANGE, RESRESTART, and RESADMINWAIT.</p> <p>To enable triggers on a specific node, add trigger keys in the following format:</p> <pre>TriggersEnabled@node1 = {RESADMINWAIT, RESNOTOFF}</pre> <p>The resadminwait trigger and resnotoff trigger are enabled on node1.</p> <p>To enable triggers on all nodes in the cluster, add trigger keys in the following format:</p> <pre>TriggersEnabled = {RESADMINWAIT, RESNOTOFF}</pre> <p>The resadminwait trigger and resnotoff trigger are enabled on all nodes.</p> <ul style="list-style-type: none">■ Type and dimension: string-keylist■ Default: {}

Resource type attributes

You can override some static attributes for resource types.

[Table C-2](#) lists the resource type attributes.

Table C-2 Resource type attributes

Resource type attributes	Description
AdvDbg (user-defined)	<p>Enables activation of advanced debugging:</p> <ul style="list-style-type: none">■ Type and dimension: string-keylist■ Default: Not applicable

Table C-2 Resource type attributes (*continued*)

Resource type attributes	Description
AgentClass (user-defined)	<p>Indicates the scheduling class for the VCS agent process.</p> <p>Use only one of the following sets of attributes to configure scheduling class and priority for VCS:</p> <ul style="list-style-type: none"> ■ AgentClass, AgentPriority, ScriptClass, and ScriptPriority Or ■ OnlineClass, OnlinePriority, EPClass, and EPPriority ■ Type and dimension: string-scalar ■ Default: TS
AgentDirectory (user-defined)	<p>Complete path of the directory in which the agent binary and scripts are located.</p> <p>If none of the above directories exist, the agent does not start.</p> <p>Use this attribute in conjunction with the AgentFile attribute to specify a different location or different binary for the agent.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-scalar ■ Default = ""
AgentFailedOn (system use only)	<p>A list of systems on which the agent for the resource type has failed.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-keylist ■ Default: Not applicable.
AgentFile (user-defined)	<p>Complete name and path of the binary for an agent. If you do not specify a value for this attribute, VCS uses the agent binary at the path defined by the AgentDirectory attribute.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-scalar ■ Default = ""
AgentPriority (user-defined)	<p>Indicates the priority in which the agent process runs.</p> <p>Use only one of the following sets of attributes to configure scheduling class and priority for VCS:</p> <ul style="list-style-type: none"> ■ AgentClass, AgentPriority, ScriptClass, and ScriptPriority Or ■ OnlineClass, OnlinePriority, EPClass, and EPPriority <p>Type and dimension: string-scalar</p> <p>Default: 0</p>

Table C-2 Resource type attributes (*continued*)

Resource type attributes	Description
AgentReplyTimeout (user-defined)	<p>The number of seconds the engine waits to receive a heartbeat from the agent before restarting the agent.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: 130 seconds
AgentStartTimeout (user-defined)	<p>The number of seconds after starting the agent that the engine waits for the initial agent "handshake" before restarting the agent.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: 60 seconds
AlertOnMonitorTimeouts (user-defined) Note: This attribute can be overridden.	<p>When a monitor times out as many times as the value or a multiple of the value specified by this attribute, then VCS sends an SNMP notification to the user. If this attribute is set to a value, say N, then after sending the notification at the first monitor timeout, VCS also sends an SNMP notification at each N-consecutive monitor timeout including the first monitor timeout for the second-time notification.</p> <p>When AlertOnMonitorTimeouts is set to 0, VCS will send an SNMP notification to the user only for the first monitor timeout; VCS will not send further notifications to the user for subsequent monitor timeouts until the monitor returns a success.</p> <p>The AlertOnMonitorTimeouts attribute can be used in conjunction with the FaultOnMonitorTimeouts attribute to control the behavior of resources of a group configured under VCS in case of monitor timeouts. When FaultOnMonitorTimeouts is set to 0 and AlertOnMonitorTimeouts is set to some value for all resources of a service group, then VCS will not perform any action on monitor timeouts for resources configured under that service group, but will only send notifications at the frequency set in the AlertOnMonitorTimeouts attribute.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: 0
ArgList (user-defined)	<p>An ordered list of attributes whose values are passed to the open, close, online, offline, monitor, clean, info, and action functions.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-vector ■ Default: Not applicable.
CleanRetryLimit (user-defined)	<p>Number of times to retry the clean function before moving a resource to ADMIN_WAIT state. If set to 0, clean is re-tried indefinitely.</p> <p>The valid values of this attribute are in the range of 0-1024.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: 0

Table C-2 Resource type attributes (*continued*)

Resource type attributes	Description
<p>ConflInterval (user-defined)</p> <p>Note: This attribute can be overridden.</p>	<p>When a resource has remained online for the specified time (in seconds), previous faults and restart attempts are ignored by the agent. (See ToleranceLimit and RestartLimit attributes for details.)</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: 600 seconds
<p>ContainerOpts (system use only)</p>	<p>Specifies information that passes to the agent that controls the resources. These values are only effective when you set the ContainerInfo service group attribute.</p> <ul style="list-style-type: none"> ■ RunInContainer When the value of the RunInContainer key is 1, the agent function (entry point) for that resource runs inside of the local container. When the value of the RunInContainer key is 0, the agent function (entry point) for that resource runs outside the local container (in the global environment). ■ PassCInfo When the value of the PassCInfo key is 1, the agent function receives the container information that is defined in the service group's ContainerInfo attribute. An example use of this value is to pass the name of the container to the agent. When the value of the PassCInfo key is 0, the agent function does not receive the container information that is defined in the service group's ContainerInfo attribute.
<p>EPClass (user-defined)</p>	<p>Enables you to control the scheduling class for the agent functions (entry points) other than the online entry point whether the entry point is in C or scripts.</p> <p>The following values are valid for this attribute:</p> <ul style="list-style-type: none"> ■ RT (Real Time) ■ TS (Time Sharing) ■ -1—indicates that VCS does not use this attribute to control the scheduling class of entry points. <p>Use only one of the following sets of attributes to configure scheduling class and priority for VCS:</p> <ul style="list-style-type: none"> ■ AgentClass, AgentPriority, ScriptClass, and ScriptPriority Or ■ OnlineClass, OnlinePriority, EPClass, and EPPriority ■ Type and dimension: string-scalar ■ Default: -1

Table C-2 Resource type attributes (*continued*)

Resource type attributes	Description
EPPriority (user-defined)	<p>Enables you to control the scheduling priority for the agent functions (entry points) other than the online entry point. The attribute controls the agent function priority whether the entry point is in C or scripts.</p> <p>The following values are valid for this attribute:</p> <ul style="list-style-type: none"> ■ 0—indicates the default priority value for the configured scheduling class as given by the EPClass attribute for the operating system. ■ Greater than 0—indicates a value greater than the default priority for the operating system. Veritas recommends a value of greater than 0 for this attribute. A system that has a higher load requires a greater value. ■ -1—indicates that VCS does not use this attribute to control the scheduling priority of entry points. <p>Use only one of the following sets of attributes to configure scheduling class and priority for VCS:</p> <ul style="list-style-type: none"> ■ AgentClass, AgentPriority, ScriptClass, and ScriptPriority Or ■ OnlineClass, OnlinePriority, EPClass, and EPPriority ■ Type and dimension: string-scalar ■ Default: -1
ExternalStateChange (user-defined) Note: This attribute can be overridden.	<p>Defines how VCS handles service group state when resources are intentionally brought online or taken offline outside of VCS control.</p> <p>The attribute can take the following values:</p> <p>OnlineGroup: If the configured application is started outside of VCS control, VCS brings the corresponding service group online.</p> <p>OfflineGroup: If the configured application is stopped outside of VCS control, VCS takes the corresponding service group offline.</p> <p>OfflineHold: If a configured application is stopped outside of VCS control, VCS sets the state of the corresponding VCS resource as offline. VCS does not take any parent resources or the service group offline.</p> <p>OfflineHold and OfflineGroup are mutually exclusive.</p>

Table C-2 Resource type attributes (*continued*)

Resource type attributes	Description
FaultOnMonitorTimeouts (user-defined) Note: This attribute can be overridden.	<p>When a monitor times out as many times as the value specified, the corresponding resource is brought down by calling the clean function. The resource is then marked FAULTED, or it is restarted, depending on the value set in the RestartLimit attribute.</p> <p>When FaultOnMonitorTimeouts 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.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: 4
FaultPropagation (user-defined) Note: This attribute can be overridden.	<p>Specifies if VCS should propagate the fault up to parent resources and take the entire service group offline when a resource faults.</p> <p>The value 1 indicates that when a resource faults, VCS fails over the service group, if the group's AutoFailOver attribute is set to 1. The value 0 indicates that when a resource faults, VCS does not take other resources offline, regardless of the value of the Critical attribute. The service group does not fail over on resource fault.</p> <ul style="list-style-type: none"> ■ Type and dimension: boolean-scalar ■ Default: 1
FireDrill (user-defined)	<p>Specifies whether or not fire drill is enabled for the resource type. If the value is:</p> <ul style="list-style-type: none"> ■ 0: Fire drill is disabled. ■ 1: Fire drill is enabled. <p>You can override this attribute.</p> <ul style="list-style-type: none"> ■ Type and dimension: boolean-scalar ■ Default: 0
InfoInterval (user-defined)	<p>Duration (in seconds) after which the info function is invoked by the agent framework for ONLINE resources of the particular resource type.</p> <p>If set to 0, the agent framework does not periodically invoke the info function. To manually invoke the info function, use the command <code>hares -refreshinfo</code>. If the value you designate is 30, for example, the function is invoked every 30 seconds for all ONLINE resources of the particular resource type.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: 0

Table C-2 Resource type attributes (*continued*)

Resource type attributes	Description
IntentionalOffline (user-defined)	<p>Defines how VCS reacts when a configured application is intentionally stopped outside of VCS control.</p> <p>Add this attribute for agents that support detection of an intentional offline outside of VCS control. Note that the intentional offline feature is available for agents registered as V51 or later.</p> <p>The value 0 instructs the agent to register a fault and initiate the failover of a service group when the supported resource is taken offline outside of VCS control.</p> <p>The value 1 instructs VCS to take the resource offline when the corresponding application is stopped outside of VCS control.</p> <ul style="list-style-type: none"> ■ Type and dimension: boolean-scalar ■ Default: 0
LogDbg (user-defined)	<p>Indicates the debug severities enabled for the resource type or agent framework. Debug severities used by the agent functions are in the range of DBG_1–DBG_21. The debug messages from the agent framework are logged with the severities DBG_AGINFO, DBG_AGDEBUG and DBG_AGTRACE, representing the least to most verbose.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-keylist ■ Default: {} (none) <p>The LogDbg attribute can be overridden. Using the LogDbg attribute, you can set DBG_AGINFO, DBG_AGTRACE, and DBG_AGDEBUG severities at the resource level, but it does not have an impact as these levels are agent-type specific. Veritas recommends to set values between DBG_1 to DBG_21 at resource level using the LogDbg attribute.</p>
LogFileSize (user-defined)	<p>Specifies the size (in bytes) of the agent log file. Minimum value is 64 KB. Maximum value is 134217728 bytes (128MB).</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: 33554432 (32 MB)
LogViaHalog (user-defined)	<p>Enables the log of all the entry points to be logged either in the respective agent log file or the engine log file based on the values configured.</p> <ul style="list-style-type: none"> ■ 0: The agent's log goes into the respective agent log file. ■ 1: The C/C++ entry point's logs goes into the agent log file and the script entry point's logs goes into the engine log file using the <code>halog</code> command. <p>Type: boolean-scalar</p> <p>Default: 0</p>

Table C-2 Resource type attributes (*continued*)

Resource type attributes	Description
MigrateWaitLimit (user-defined)	<p>Number of monitor intervals to wait for a resource to migrate after the migrating procedure is complete. MigrateWaitLimit is applicable for the source and target node because the migrate operation takes the resource offline on the source node and brings the resource online on the target node. You can also define MigrateWaitLimit as the number of monitor intervals to wait for the resource to go offline on the source node after completing the migrate procedure and the number of monitor intervals to wait for the resource to come online on the target node after resource is offline on the source node.</p> <ul style="list-style-type: none">■ Type and dimension: integer-scalar■ Default: 2 <p>Note: This attribute can be overridden.</p> <p>Probes fired manually are counted when MigrateWaitLimit is set and the resource is waiting to migrate. For example, if the MigrateWaitLimit of a resource is set to 5 and the MonitorInterval is set to 60 (seconds), the resource waits for a maximum of five monitor intervals (that is, 5 x 60), and if all five monitors within MigrateWaitLimit report the resource as online on source node, it sets the ADMIN_WAIT flag. If you run another probe, the resource waits for four monitor intervals (that is, 4 x 60), and if the fourth monitor does not report the state as offline on source, it sets the ADMIN_WAIT flag. This procedure is repeated for 5 complete cycles. Similarly, if resource not moved to online state within the MigrateWaitLimit then it sets the ADMIN_WAIT flag.</p>
MigrateTimeout (user-defined)	<p>Maximum time (in seconds) within which the migrate procedure must complete or else be terminated.</p> <ul style="list-style-type: none">■ Type and dimension: integer-scalar■ Default: 600 seconds <p>Note: This attribute can be overridden.</p>
MonitorInterval (user-defined) Note: This attribute can be overridden.	<p>Duration (in seconds) between two consecutive monitor calls for an ONLINE or transitioning resource.</p> <p>A low value may impact performance if many resources of the same type exist. A high value may delay detection of a faulted resource.</p> <ul style="list-style-type: none">■ Type and dimension: integer-scalar■ Default: 60 seconds

Table C-2 Resource type attributes (*continued*)

Resource type attributes	Description
MonitorStatsParam (user-defined)	<p>Stores the required parameter values for calculating monitor time statistics.</p> <pre>static str MonitorStatsParam = {Frequency = 10, ExpectedValue = 3000, ValueThreshold = 100, AvgThreshold = 40}</pre> <p>Frequency: The number of monitor cycles after which the average monitor cycle time should be computed and sent to the engine. If configured, the value for this attribute must be between 1 and 30. The value 0 indicates that the monitor cycle time should not be computed. Default=0.</p> <p>ExpectedValue: The expected monitor time in milliseconds for all resources of this type. Default=100.</p> <p>ValueThreshold: The acceptable percentage difference between the expected monitor cycle time (ExpectedValue) and the actual monitor cycle time. Default=100.</p> <p>AvgThreshold: The acceptable percentage difference between the benchmark average and the moving average of monitor cycle times. Default=40.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-association ■ Default: Different value for each parameter.
NumThreads (user-defined)	<p>Number of threads used within the agent process for managing resources. This number does not include threads used for other internal purposes.</p> <p>If the number of resources being managed by the agent is less than or equal to the NumThreads value, only that many number of threads are created in the agent. Addition of more resources does not create more service threads. Similarly deletion of resources causes service threads to exit. Thus, setting NumThreads to 1 forces the agent to just use 1 service thread no matter what the resource count is. The agent framework limits the value of this attribute to 30.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: 10
OfflineMonitorInterval (user-defined) Note: This attribute can be overridden.	<p>Duration (in seconds) between two consecutive monitor calls for an OFFLINE resource. If set to 0, OFFLINE resources are not monitored.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: 300 seconds

Table C-2 Resource type attributes (*continued*)

Resource type attributes	Description
OfflineWaitLimit (user-defined) Note: This attribute can be overridden.	<p>Number of monitor intervals to wait for the resource to go offline after completing the offline procedure. Increase the value of this attribute if the resource is likely to take a longer time to go offline.</p> <p>Probes fired manually are counted when OfflineWaitLimit is set and the resource is waiting to go offline. For example, say the OfflineWaitLimit of a resource is set to 5 and the MonitorInterval is set to 60. The resource waits for a maximum of five monitor intervals (five times 60), and if all five monitors within OfflineWaitLimit report the resource as online, it calls the clean agent function. If the user fires a probe, the resource waits for four monitor intervals (four times 60), and if the fourth monitor does not report the state as offline, it calls the clean agent function. If the user fires another probe, one more monitor cycle is consumed and the resource waits for three monitor intervals (three times 60), and if the third monitor does not report the state as offline, it calls the clean agent function.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: 0
OnlineClass (user-defined)	<p>Enables you to control the scheduling class for the online agent function (entry point). This attribute controls the class whether the entry point is in C or scripts.</p> <p>The following values are valid for this attribute:</p> <ul style="list-style-type: none"> ■ RT (Real Time) ■ TS (Time Sharing) ■ -1—indicates that VCS does not use this attribute to control the scheduling class of entry points. <p>Use only one of the following sets of attributes to configure scheduling class and priority for VCS:</p> <ul style="list-style-type: none"> ■ AgentClass, AgentPriority, ScriptClass, and ScriptPriority Or ■ OnlineClass, OnlinePriority, EPClass, and EPPriority ■ Type and dimension: string-scalar ■ Default: -1

Table C-2 Resource type attributes (*continued*)

Resource type attributes	Description
OnlinePriority (user-defined)	<p>Enables you to control the scheduling priority for the online agent function (entry point). This attribute controls the priority whether the entry point is in C or scripts.</p> <p>The following values are valid for this attribute:</p> <ul style="list-style-type: none"> 0—indicates the default priority value for the configured scheduling class as given by the OnlineClass for the operating system. Veritas recommends that you set the value of the OnlinePriority attribute to 0. Greater than 0—indicates a value greater than the default priority for the operating system. -1—indicates that VCS does not use this attribute to control the scheduling priority of entry points. <p>Use only one of the following sets of attributes to configure scheduling class and priority for VCS:</p> <ul style="list-style-type: none"> AgentClass, AgentPriority, ScriptClass, and ScriptPriority Or OnlineClass, OnlinePriority, EPClass, and EPPriority <p>Type and dimension: string-scalar</p> <p>Default: -1</p>
OnlineRetryLimit (user-defined) Note: This attribute can be overridden.	<p>Number of times to retry the <code>online</code> operation if the attempt to online a resource is unsuccessful. This parameter is meaningful only if the clean operation is implemented.</p> <ul style="list-style-type: none"> Type and dimension: integer-scalar Default: 0
OnlineWaitLimit (user-defined) Note: This attribute can be overridden.	<p>Number of monitor intervals to wait for the resource to come online after completing the online procedure. Increase the value of this attribute if the resource is likely to take a longer time to come online.</p> <p>Each probe command fired from the user is considered as one monitor interval. For example, say the OnlineWaitLimit of a resource is set to 5. This means that the resource will be moved to a faulted state after five monitor intervals. If the user fires a probe, then the resource will be faulted after four monitor cycles, if the fourth monitor does not report the state as ONLINE. If the user again fires a probe, then one more monitor cycle is consumed and the resource will be faulted if the third monitor does not report the state as ONLINE.</p> <ul style="list-style-type: none"> Type and dimension: integer-scalar Default: 2

Table C-2 Resource type attributes (*continued*)

Resource type attributes	Description
Operations (user-defined)	<p>Indicates valid operations for resources of the resource type. Values are OnOnly (can online only), OnOff (can online and offline), None (cannot online or offline).</p> <ul style="list-style-type: none"> ■ Type and dimension: string-scalar ■ Default: OnOff
RestartLimit (user-defined) Note: This attribute can be overridden.	<p>Number of times to retry bringing a resource online when it is taken offline unexpectedly and before VCS declares it FAULTED.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: 0
ScriptClass (user-defined)	<p>Indicates the scheduling class of the script processes (for example, online) created by the agent.</p> <p>Use only one of the following sets of attributes to configure scheduling class and priority for VCS:</p> <ul style="list-style-type: none"> ■ AgentClass, AgentPriority, ScriptClass, and ScriptPriority Or ■ OnlineClass, OnlinePriority, EPClass, and EPPriority ■ Type and dimension: string-scalar ■ Default: -1 ■ Type and dimension: string-scalar ■ Default: TS
ScriptPriority (user-defined)	<p>Indicates the priority of the script processes created by the agent.</p> <p>Use only one of the following sets of attributes to configure scheduling class and priority for VCS:</p> <ul style="list-style-type: none"> ■ AgentClass, AgentPriority, ScriptClass, and ScriptPriority Or ■ OnlineClass, OnlinePriority, EPClass, and EPPriority ■ Type and dimension: string-scalar ■ Default: 0
SourceFile (user-defined)	<p>File from which the configuration is read. Do not configure this attribute in main.cf.</p> <p>Make sure the path exists on all nodes before running a command that configures this attribute.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-scalar ■ Default: .types.cf

Table C-2 Resource type attributes (*continued*)

Resource type attributes	Description
SupportedActions (user-defined)	Valid action tokens for the resource type. <ul style="list-style-type: none">■ Type and dimension: string-vector■ Default: {}
SupportedOperations (user-defined)	Indicates the additional operations for a resource type or an agent. Only <code>migrate</code> keyword is supported. <ul style="list-style-type: none">■ Type and dimension: string-keylist■ Default: {}
ToleranceLimit (user-defined) Note: This attribute can be overridden.	After a resource goes online, the number of times the monitor function should return OFFLINE before declaring the resource FAULTED. A large value could delay detection of a genuinely faulted resource. <ul style="list-style-type: none">■ Type and dimension: integer-scalar■ Default: 0
TypeOwner (user-defined)	This attribute is used for VCS notification. VCS sends notifications to persons designated in this attribute when an event occurs related to the agent's resource type. If the agent of that type faults or restarts, VCS send notification to the TypeOwner. Note that while VCS logs most events, not all events trigger notifications. Make sure to set the severity level at which you want notifications to be sent to TypeOwner or to at least one recipient defined in the Smtprcipients attribute of the NotifierMngr agent. <ul style="list-style-type: none">■ Type and dimension: string-scalar■ Default: ""■ Example: "jdoe@example.com"
TypeRecipients (user-defined)	The email-ids set in the TypeRecipients attribute receive email notification for events related to a specific agent. There are only two types of events related to an agent for which notifications are sent: <ul style="list-style-type: none">■ Agent fault - Warning■ Agent restart - Information

Service group attributes

Table C-3 lists the service group attributes.

Table C-3 Service group attributes

Service Group Attributes	Definition
ActiveCount (system use only)	<p>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.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: Not applicable.
AdministratorGroups (user-defined)	<p>List of operating system user account groups that have administrative privileges on the service group.</p> <p>This attribute applies to clusters running in secure mode.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-keylist ■ Default: {} (none)
Administrators (user-defined)	<p>List of VCS users with privileges to administer the group.</p> <p>Note: A Group Administrator can perform all operations related to a specific service group, but cannot perform generic cluster operations.</p> <p>See “About VCS user privileges and roles” on page 66.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-keylist ■ Default: {} (none)
Authority (user-defined)	<p>Indicates whether or not the local cluster is allowed to bring the service group online. If set to 0, it is not, if set to 1, it is. Only one cluster can have this attribute set to 1 for a specific global group.</p> <p>See “About serialization–The Authority attribute” on page 391.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: 0
AutoClearCount (System use only)	<p>Indicates the number of attempts that the VCS engine made to clear the state of the service group that has faulted and does not have a failover target. This attribute is used only if the AutoClearLimit attribute is set for the service group.</p>
AutoClearInterval (user-defined)	<p>Indicates the interval in seconds after which a service group that has faulted and has no failover target is cleared automatically. The state of the service group is cleared only if AutoClearLimit is set to a non-zero value.</p> <p>Default: 0</p>

Table C-3 Service group attributes (*continued*)

Service Group Attributes	Definition
AutoclearLimit (user-defined)	Defines the number of attempts to be made to clear the Faulted state of a service group. Disables the auto-clear feature when set to zero.
AutoDisabled (system use only)	<p>Indicates that VCS does not know the status of a service group (or specified system for parallel service groups). This could occur because the group is not probed (on specified system for parallel groups) in the SystemList attribute. Or the VCS engine is not running on a node designated in the SystemList attribute, but the node is visible.</p> <p>When VCS does not know the status of a service group on a node but you want VCS to consider the service group enabled, perform this command to change the AutoDisabled value to 0.</p> <pre>hagrp -autoenable grp -sys sys1</pre> <p>This command instructs VCS that even though VCS has marked the service group auto-disabled, you are sure that the service group is not online on sys1. For failover service groups, this is important because the service groups now can be brought online on remaining nodes.</p> <ul style="list-style-type: none">■ Type and dimension: boolean-scalar■ Default: 0
AutoFailOver (user-defined)	<p>Indicates whether VCS initiates an automatic failover if the service group faults.</p> <p>The attribute can take the following values:</p> <ul style="list-style-type: none">■ 0—VCS does not fail over the service group.■ 1—VCS automatically fails over the service group if a suitable node exists for failover. <p>See “About controlling failover on service group or system faults” on page 294.</p> <ul style="list-style-type: none">■ Type and dimension: boolean-scalar■ Default: 1 (enabled)

Table C-3 Service group attributes (*continued*)

Service Group Attributes	Definition
AutoRestart (user-defined)	<p>Restarts a service group after a faulted persistent resource becomes online.</p> <p>The attribute can take the following values:</p> <ul style="list-style-type: none">■ 0—Autorestart is disabled.■ 1—Autorestart is enabled.■ 2—When a faulted persistent resource recovers from a fault, the VCS engine clears the faults on all non-persistent faulted resources on the system. It then restarts the service group. <p>See “About service group dependencies” on page 335.</p> <p>Note: This attribute applies only to service groups containing persistent resources.</p> <ul style="list-style-type: none">■ Type and dimension: integer-scalar■ Default: 1 (enabled)
AutoStart (user-defined)	<p>Designates whether a service group is automatically started when VCS is started.</p> <ul style="list-style-type: none">■ Type and dimension: boolean-scalar■ Default: 1 (enabled)
AutoStartIfPartial (user-defined)	<p>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.</p> <ul style="list-style-type: none">■ Type and dimension: boolean-scalar■ Default: 1 (enabled)

Table C-3 Service group attributes (*continued*)

Service Group Attributes	Definition
AutoStartList (user-defined)	<p>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 the service group is not already running on another system in the cluster, the group is brought online when the system is started.</p> <p>VCS uses the AutoStartPolicy attribute to determine the system on which to bring the service group online.</p> <p>Note: 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.</p> <ul style="list-style-type: none">■ Type and dimension: string-keylist■ Default: {} (none)
AutoStartPolicy (user-defined)	<p>Sets the policy VCS uses to determine the system on which a service group is brought online during an autostart operation if multiple systems exist.</p> <p>This attribute has three options:</p> <p>Order (default)—Systems are chosen in the order in which they are defined in the AutoStartList attribute.</p> <p>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.</p> <p>Note: You cannot set the value Load when the cluster attribute Statistics is set to Enabled.</p> <p>Priority—Systems are chosen in the order of their priority in the SystemList attribute. Systems with the lowest priority is chosen first.</p> <ul style="list-style-type: none">■ Type and dimension: string-scalar■ Default: Order

Table C-3 Service group attributes (*continued*)

Service Group Attributes	Definition
CapacityReserved (system use only)	<p>Indicates whether capacity is reserved to bring service groups online or to fail them over. Capacity is reserved only when the service group attribute FailOverPolicy is set to BiggestAvailable.</p> <p>This attribute is localized.</p> <ul style="list-style-type: none"> ■ Type and dimension: Boolean ■ Default: "" <p>Possible values:</p> <p>1: Capacity is reserved.</p> <p>0: Capacity is not reserved.</p> <p>The value can be reset using the <code>hagrp -flush</code> command.</p> <p>To list this attribute, use the <code>-all</code> option with the <code>hagrp -display</code> command.</p>
ClusterFailOverPolicy (user-defined)	<p>Determines how a global service group behaves when a cluster faults or when a global group faults. The attribute can take the following values:</p> <p>Manual—The group does not fail over to another cluster automatically.</p> <p>Auto—The group fails over to another cluster automatically if it is unable to fail over within the local cluster, or if the entire cluster faults.</p> <p>Connected—The group fails over automatically to another cluster only if it is unable to fail over within the local cluster.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-scalar ■ Default: Manual
ClusterList (user-defined)	<p>Specifies the list of clusters on which the service group is configured to run.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-association ■ Default: {} (none)
CurrentCount (system use only)	<p>Number of systems on which the service group is active.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: Not applicable.

Table C-3 Service group attributes (*continued*)

Service Group Attributes	Definition
DisableFaultMessages (user-defined)	<p>Suppresses fault and failover messages, for a group and its resources, from getting logged in the VCS engine log file. This attribute does not suppress the information messages getting logged in the log file.</p> <p>The attribute can take the following values:</p> <ul style="list-style-type: none"> ■ 0 - Logs all the fault and failover messages for the service group and its resources. ■ 1 - Disables the fault and failover messages of the service groups, but continues to log resource messages. ■ 2 - Disables the fault and failover messages of the service group resources, but continues to log service group messages. ■ 3 - Disables the fault and failover messages of both service groups and its resources.
DeferAutoStart (system use only)	<p>Indicates whether HAD defers the auto-start of a global group in the local cluster in case the global cluster is not fully connected.</p> <ul style="list-style-type: none"> ■ Type and dimension: boolean-scalar ■ Default: Not applicable
Enabled (user-defined)	<p>Indicates if a service group can be failed over or brought online.</p> <p>The attribute can have global or local scope. If you define local (system-specific) scope for this attribute, VCS prevents the service group from coming online on specified systems that have a value of 0 for the attribute. You can use this attribute to prevent failovers on a system when performing maintenance on the system.</p> <ul style="list-style-type: none"> ■ Type and dimension: boolean-scalar ■ Default: 1 (enabled)
Evacuate (user-defined)	<p>Indicates if VCS initiates an automatic failover when user issues <code>hastop -local -evacuate</code>.</p> <ul style="list-style-type: none"> ■ Type and dimension: boolean-scalar ■ Default: 1
Evacuating (system use only)	<p>Indicates the node ID from which the service group is being evacuated.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: Not applicable

Table C-3 Service group attributes (*continued*)

Service Group Attributes	Definition
EvacList (system use only)	<p>Contains list of pairs of low priority service groups and the systems on which they will be evacuated.</p> <p>For example:</p> <pre>Grp1 EvacList grp2 Sys0 grp3 Sys0 grp4 Sys4</pre> <p>Type and dimension: string-association</p> <p>Default: Not applicable.</p>
EvacuatingForGroup (system use only)	<p>Displays the name of the high priority service group for which evacuation is in progress. The service group name is visible only as long as the evacuation is in progress.</p> <p>Type and dimension: string-scalar</p> <p>Default: Not applicable.</p>
Failover (system use only)	<p>Indicates service group is in the process of failing over.</p> <ul style="list-style-type: none">■ Type and dimension: boolean-scalar■ Default: Not applicable

Table C-3 Service group attributes (*continued*)

Service Group Attributes	Definition
FailOverPolicy (user-defined)	<p>Defines the failover policy used by VCS to determine the system to which a group fails over. It is also used to determine the system on which a service group has been brought online through manual operation.</p> <p>The policy is defined only for clusters that contain multiple systems:</p> <p>Priority—The system defined as the lowest priority in the SystemList attribute is chosen.</p> <p>Load—The system with the highest value of AvailableCapacity is chosen.</p> <p>RoundRobin—Systems are chosen based on the active service groups they host. The system with the least number of active service groups is chosen first.</p> <p>BiggestAvailable—Systems are chosen based on the forecasted available capacity for all systems in the SystemList. The system with the highest available capacity forecasted is selected.</p> <p>Note: VCS selects the node in an alphabetical order when VCS detects two systems with same values set for the policy Priority, Load, RoundRobin, or BiggestAvailable.</p> <p>Prerequisites for setting FailOverPolicy to BiggestAvailable:</p> <ul style="list-style-type: none">■ The cluster attribute Statistics must be set to Enabled.■ Veritas recommends that the cluster attribute HostMeters should contain at least one key.■ The service group attribute Load must contain at least one key.■ You cannot change the attribute from BiggestAvailable to some other value, when the service group attribute CapacityReserved is set to 1 because the VCS engine reserves system capacity when it determines BiggestAvailable for the service group. When the service group online transition completes and after the next forecast cycle, CapacityReserved is reset.■ Type and dimension: string-scalar■ Default: Priority

Table C-3 Service group attributes (*continued*)

Service Group Attributes	Definition
FaultPropagation (user-defined)	<p>Specifies if VCS should propagate the fault up to parent resources and take the entire service group offline when a resource faults.</p> <p>The value 1 indicates that when a resource faults, VCS fails over the service group, if the group's AutoFailOver attribute is set to 1. If The value 0 indicates that when a resource faults, VCS does not take other resources offline, regardless of the value of the Critical attribute. The service group does not fail over on resource fault.</p> <ul style="list-style-type: none"> ■ Type and dimension: boolean-scalar ■ Default: 1
FromQ (system use only)	<p>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.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-association ■ Default: Not applicable
Frozen (system use only)	<p>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.)</p> <ul style="list-style-type: none"> ■ Type and dimension: boolean-scalar ■ Default: 0 (not frozen)
GroupOwner (user-defined)	<p>This attribute is used for VCS email notification and logging. VCS sends email notification to the person designated in this attribute when events occur that are related to the service group. Note that while VCS logs most events, not all events trigger notifications.</p> <p>Make sure to set the severity level at which you want notifications to be sent to GroupOwner or to at least one recipient defined in the SmtprRecipients attribute of the NotifierMngr agent.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-scalar ■ Default: ""

Table C-3 Service group attributes (*continued*)

Service Group Attributes	Definition
GroupRecipients (user-defined)	<p>This attribute is used for VCS email notification. VCS sends email notification to persons designated in this attribute when events related to the service group occur and when the event's severity level is equal to or greater than the level specified in the attribute.</p> <p>Make sure to set the severity level at which you want notifications to be sent to GroupRecipients or to at least one recipient defined in the Smtprcipients attribute of the NotifierMngr agent.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-association ■ email id: The email address of the person registered as a recipient for notification. severity: The minimum level of severity at which notifications must be sent.
Guests (user-defined)	<p>List of operating system user accounts that have Guest privileges on the service group.</p> <p>This attribute applies to clusters running in secure mode.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-keylist ■ Default: ""
IntentOnline (system use only)	<p>Indicates whether to keep service groups online or offline.</p> <p>VCS sets this attribute to 1 if an attempt has been made to bring the service group online.</p> <p>For failover groups, VCS sets this attribute to 0 when the group is taken offline.</p> <p>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> <p>VCS sets this attribute to 2 for service groups if VCS attempts to autostart a service group; for example, attempting to bring a service group online on a system from AutoStartList.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: Not applicable.

Table C-3 Service group attributes (*continued*)

Service Group Attributes	Definition
IntentionalOnlineList (system use only)	<p>Lists the nodes where a resource that can be intentionally brought online is found ONLINE at first probe. IntentionalOnlineList is used along with AutoStartList to determine the node on which the service group should go online when a cluster starts.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-keylist ■ Default: Not applicable
LastSuccess (system use only)	<p>Indicates the time when service group was last brought online.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: Not applicable
Load (user-defined)	<p>Indicates the multidimensional value expressing load exerted by a service group on the system.</p> <p>When the cluster attribute Statistics is not enabled, the allowed key value is Units.</p> <ul style="list-style-type: none"> ■ Type and dimension: float-association ■ Default: Not applicable <p>The following additional considerations apply:</p> <ul style="list-style-type: none"> ■ You cannot change this attribute when the service group attribute CapacityReserved is set to 1 in the cluster and when the FailOverPolicy is set to BiggestAvailable. This is because the VCS engine reserves system capacity based on the service group attribute Load. <p>When the service group's online transition completes and after the next forecast cycle, CapacityReserved is reset.</p>
ManageFaults (user-defined)	<p>Specifies if VCS manages resource failures within the service group by calling the Clean function for the resources. This attribute can take the following values.</p> <p>NONE—VCS does not call the Clean function for any resource in the group. You must manually handle resource faults.</p> <p>See “About controlling Clean behavior on resource faults” on page 297.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-scalar ■ Default: ALL

Table C-3 Service group attributes (*continued*)

Service Group Attributes	Definition
ManualOps (user-defined)	<p>Indicates if manual operations are allowed on the service group.</p> <ul style="list-style-type: none"> ■ Type and dimension: boolean-scalar ■ Default = 1 (enabled)
MeterWeight (user-defined)	<p>Represents the weight given for the cluster attribute's HostMeters key to determine a target system for a service group when more than one system meets the group attribute's Load requirements.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-association ■ Default: Not applicable <p>Additional considerations for configuring this attribute in main.cf or changing it at run time:</p> <ul style="list-style-type: none"> ■ This is an optional service group attribute. If it is not defined for a group, the VCS considers the cluster attribute MeterWeight. ■ To override this attribute at an individual group level, define it at run time or in the main.cf file. Ensure that keys are subsets of the cluster attribute HostAvailableMeters. ■ You cannot change this attribute when the service group attribute CapacityReserved is set to 1 in the cluster. ■ The values for the keys represent weights of the corresponding parameters. It should be in range of 0 to 10.
MigrateQ (system use only)	<p>Indicates the system from which the service group is migrating. This attribute is specified when group failover is an indirect consequence (in situations such as a system shutdown or another group faults and is linked to this group).</p> <ul style="list-style-type: none"> ■ Type and dimension: string-association ■ Default: Not applicable
NumRetries (system use only)	<p>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.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: Not applicable

Table C-3 Service group attributes (*continued*)

Service Group Attributes	Definition
OnlineAtUnfreeze (system use only)	<p>When a node or a service group is frozen, the OnlineAtUnfreeze attribute specifies how an offline service group reacts after it or a node is unfrozen.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: Not applicable
OnlineClearParent	<p>When this attribute is enabled for a service group and the service group comes online or is detected online, VCS clears the faults on all online type parent groups, such as online local, online global, and online remote.</p> <ul style="list-style-type: none"> ■ Type and dimension: boolean-scalar ■ Default: 0 <p>For example, assume that both the parent group and the child group faulted and both cannot failover. Later, when VCS tries again to bring the child group online and the group is brought online or detected online, the VCS engine clears the faults on the parent group, allowing VCS to restart the parent group too.</p>
OnlineRetryInterval (user-defined)	<p>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.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: 0
OnlineRetryLimit (user-defined)	<p>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.</p> <p>If this attribute is set for a service group, then set a non-zero value for the OnlineRetryInterval attribute. This ensures that the service group does not fault and restart on the same node.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: 0

Table C-3 Service group attributes (*continued*)

Service Group Attributes	Definition
OperatorGroups (user-defined)	<p>List of operating system user groups that have Operator privileges on the service group. This attribute applies to clusters running in secure mode.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-keylist ■ Default: ""
Operators (user-defined)	<p>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.</p> <p>See “About VCS user privileges and roles” on page 66.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-keylist ■ Default: ""
Parallel (user-defined)	<p>Indicates if service group is failover (0), parallel (1), or hybrid(2).</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: 0
PathCount (system use only)	<p>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.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: Not applicable
PolicyIntention (system use only)	<p>Functions as a lock on service groups listed in the <code>hagrp -online -propagate</code> command and <code>hagrp -offline -propagate</code> command:</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar <p>When PolicyIntention is set to a non-zero value for the service groups in dependency tree, this attribute protects the service groups from any other operation. PolicyIntention can take three values.</p> <ul style="list-style-type: none"> ■ The value 0 indicates that the service group is not part of the <code>hagrp -online -propagate</code> operation or the <code>hagrp -offline -propagate</code> operation. ■ The value 1 indicates that the service group is part of the <code>hagrp -online -propagate</code> operation. ■ The value 2 indicates that the service group is part of the <code>hagrp -offline -propagate</code> operation.

Table C-3 Service group attributes (*continued*)

Service Group Attributes	Definition
PreOnline (user-defined)	<p>Indicates that the VCS engine should not bring online a service group in response to a manual group online, group autostart, or group failover. The engine should instead run the PreOnline trigger.</p> <p>You can set a local (per-system) value or a global value for this attribute. A per-system value enables you to control the firing of PreOnline triggers on specific nodes in the cluster.</p> <ul style="list-style-type: none"> ■ Type and dimension: boolean-scalar ■ Default: 0 <p>You can change the attribute scope from local to global as follows:</p> <pre># hagr -local <service_group_name> <attribute_name></pre> <p>You can change the attribute scope from global to local as follows:</p> <pre># hagr -global <service_group_name> <attribute_name> <value> ... <key> ... {<key> <value>} ...</pre> <p>For more information about the <code>-local</code> option and the <code>-global</code> option, see the man pages associated with the <code>hagr</code> command.</p>
PreOnlining (system use only)	<p>Indicates that VCS engine invoked the preonline script; however, the script has not yet returned with group online.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: Not applicable
PreonlineTimeout (user-defined)	<p>Defines the maximum amount of time in seconds the preonline script takes to run the command <code>hagr -online -nopre</code> for the group. Note that HAD uses this timeout during evacuation only. For example, when a user runs the command <code>hastop -local -evacuate</code> and the Preonline trigger is invoked on the system on which the service groups are being evacuated.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: 300

Table C-3 Service group attributes (*continued*)

Service Group Attributes	Definition
Prerequisites (user-defined)	<p>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:</p> <p>Prerequisites() = {Name=Value, name2=value2}.</p> <p>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().</p> <p>See System limits and service group prerequisites on page 317.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-association
PrintTree (user-defined)	<p>Indicates whether or not the resource dependency tree is written to the configuration file. The value 1 indicates the tree is written.</p> <p>Note: For very large configurations, the time taken to print the tree and to dump the configuration is high.</p> <ul style="list-style-type: none"> ■ Type and dimension: boolean-scalar ■ Default: 1
Priority (user-defined)	<p>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.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: 0
Probed (system use only)	<p>Indicates whether all enabled resources in the group have been detected by their respective agents.</p> <ul style="list-style-type: none"> ■ Type and dimension: boolean-scalar ■ Default: Not applicable
ProbesPending (system use only)	<p>The number of resources that remain to be detected by the agent on each system.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: Not applicable

Table C-3 Service group attributes (*continued*)

Service Group Attributes	Definition
Responding (system use only)	<p>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.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: Not applicable
Restart (system use only)	<p>For internal use only.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: Not applicable
SourceFile (user-defined)	<p>File from which the configuration is read. Do not configure this attribute in main.cf.</p> <p>Make sure the path exists on all nodes before running a command that configures this attribute.</p> <p>Make sure the path exists on all nodes before configuring this attribute.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-scalar ■ Default: .main.cf

Table C-3 Service group attributes (*continued*)

Service Group Attributes	Definition
<p>State (system use only)</p>	<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>MIGRATING— Group is attempting to migrate a resource from the source system to the target system. This state should be seen only as a combination of multiple states such as, ONLINE STOPPING MIGRATING, OFFLINE STARTING MIGRATING, and OFFLINE MIGRATING.</p> <p>A group state may be a combination of multiple states described above. For example, OFFLINE FAULTED, OFFLINE STARTING, PARTIAL FAULTED, PARTIAL STARTING, PARTIAL STOPPING, ONLINE STOPPING</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: Not applicable.

Table C-3 Service group attributes (*continued*)

Service Group Attributes	Definition
SysDownPolicy (user-defined)	<p>Determines whether a service group is autodisabled when the system is down and if the service group is taken offline when the system is rebooted or is shut down gracefully.</p> <p>If SysDownPolicy contains the key AutoDisableNoOffline, the following conditions apply:</p> <ul style="list-style-type: none"> ■ The service group is autodisabled when system is down, gracefully shut down, or is detected as down. ■ The service group is not taken offline when the system reboots or shuts down gracefully. <p>Valid values: Empty keylist or the key AutoDisableNoOffline</p> <p>Default: Empty keylist</p> <p>For example, if a service group with SysDownPolicy = AutoDisableNoOffline is online on system <i>sys1</i>, it has the following effect for various commands:</p> <ul style="list-style-type: none"> ■ The <code>hastop -local -evacuate</code> command for <i>sys1</i> is rejected ■ The <code>hastop -sysoffline</code> command is accepted but the service group with SysDownPolicy = AutoDisableNoOffline is not taken offline. ■ The <code>hastop -all</code> command is rejected.
SystemList (user-defined)	<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>Note: You must define this attribute prior to setting the AutoStartList attribute.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-association ■ Default: "" (none)

Table C-3 Service group attributes (*continued*)

Service Group Attributes	Definition
SystemZones (user-defined)	<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> <ul style="list-style-type: none"> ■ Type and dimension: integer-association ■ Default: "" (none) <p>Note: You cannot modify this attribute when SiteAware is set as 1 and Sites are defined.</p>
Tag (user-defined)	<p>Identifies special-purpose service groups created for specific VCS products.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-scalar ■ Default: Not applicable.
TargetCount (system use only)	<p>Indicates the number of target systems on which the service group should be brought online.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: Not applicable.
TFrozen (user-defined)	<p>Indicates if service groups can be brought online or taken offline on nodes in the cluster. Service groups cannot be brought online or taken offline if the value of the attribute is 1.</p> <ul style="list-style-type: none"> ■ Type and dimension: boolean-scalar ■ Default: 0 (not frozen)
ToQ (system use only)	<p>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.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-association ■ Default: Not applicable

Table C-3 Service group attributes (*continued*)

Service Group Attributes	Definition
TriggerEvent (user-defined)	<p>For internal use only.</p> <ul style="list-style-type: none"> ■ Type and dimension: boolean-scalar ■ Default: Not applicable
TriggerPath (user-defined)	<p>Enables you to customize the trigger path.</p> <p>If a trigger is enabled but the trigger path is "" (default), VCS invokes the trigger from the \$VCS_HOME/bin/<i>triggers</i> directory. If you specify an alternate directory, VCS invokes the trigger from that path. The value is case-sensitive. VCS does not trim the leading spaces or trailing spaces in the Trigger Path value. If the path contains leading spaces or trailing spaces, the trigger might fail to get executed.</p> <p>The path that you specify must be in the following format:</p> <p><code>\$VCS_HOME/TriggerPath/Trigger</code></p> <p>For example, if TriggerPath is set to mytriggers/sg1, VCS looks for the preonline trigger scripts in the \$VCS_HOME/mytriggers/sg1/preonline/ directory.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-scalar ■ Default: ""
TriggerResFault (user-defined)	<p>Defines whether VCS invokes the resfault trigger when a resource faults. The value 0 indicates that VCS does not invoke the trigger.</p> <ul style="list-style-type: none"> ■ Type and dimension: boolean-scalar ■ Default: 1
TriggerResRestart (user-defined)	<p>Determines whether or not to invoke the resrestart trigger if resource restarts.</p> <p>See “About the resrestart event trigger” on page 381.</p> <p>To invoke the resrestart trigger for a specific resource, enable this attribute at the resource level.</p> <p>See “Resource attributes” on page 525.</p> <ul style="list-style-type: none"> ■ Type and dimension: boolean-scalar ■ Default: 0 (disabled)

Table C-3 Service group attributes (*continued*)

Service Group Attributes	Definition
TriggerResStateChange (user-defined)	<p>Determines whether or not to invoke the resstatechange trigger if resource state changes.</p> <p>See “About the resstatechange event trigger” on page 381.</p> <p>To invoke the resstatechange trigger for a specific resource, enable this attribute at the resource level.</p> <p>See “Resource attributes” on page 525.</p> <ul style="list-style-type: none"> ■ Type and dimension: boolean-scalar ■ Default: 0 (disabled)
TriggersEnabled (user-defined)	<p>Determines if a specific trigger is enabled or not.</p> <p>Triggers are disabled by default. You can enable specific triggers on all nodes or on selected nodes. Valid values are VIOLATION, NOFAILOVER, PREONLINE, POSTONLINE, POSTOFFLINE, RESFAULT, RESSTATECHANGE, and RESRESTART.</p> <p>To enable triggers on a node, add trigger keys in the following format:</p> <p>TriggersEnabled@node1 = {POSTOFFLINE, POSTONLINE}</p> <p>The postoffline trigger and postonline trigger are enabled on node1.</p> <p>To enable triggers on all nodes in the cluster, add trigger keys in the following format:</p> <p>TriggersEnabled = {POSTOFFLINE, POSTONLINE}</p> <p>The postoffline trigger and postonline trigger are enabled on all nodes.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-keylist ■ Default: {} <p>You can change the attribute scope from local to global as follows:</p> <pre># hagrps -local <service_group_name> <attribute_name></pre> <p>You can change the attribute scope from global to local as follows:</p> <pre># hagrps -global <service_group_name> <attribute_name> <value> ... <key> ... {<key> <value>} ...</pre> <p>For more information about the <code>-local</code> option and the <code>-global</code> option, see the man pages associated with the <code>hagrps</code> command.</p>

Table C-3 Service group attributes (*continued*)

Service Group Attributes	Definition
TypeDependencies (user-defined)	<p>Creates a dependency (via an ordered list) between resource types specified in the service group list, and all instances of the respective resource type.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-keylist ■ Default: ""
UnSteadyCount (system use only)	<p>Represents the total number of resources with pending online or offline operations. This is a localized attribute.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer ■ Default: 0 <p>To list this attribute, use the <code>-all</code> option with the <code>hagrp -display</code> command.</p> <p>The <code>hagrp -flush</code> command resets this attribute.</p>
UserAssoc (user-defined)	<p>This is a free form string-association attribute to hold any key-value pair. "Name" and "UTimeout" keys are reserved by VCS health view. You must not delete these keys or update the values corresponding to these keys, but you can add other keys and use it for any other purpose.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-association ■ Default: {} <p>You can change the attribute scope from local to global as follows:</p> <pre># hagrp -local <service_group_name> <attribute_name></pre> <p>You can change the attribute scope from global to local as follows:</p> <pre># hagrp -global <service_group_name> <attribute_name> <value> ... <key> ... {<key> <value>} ...</pre> <p>For more information about the <code>-local</code> option and <code>-global</code> option, see the man pages associated with the <code>hagrp</code> command.</p>
UserIntGlobal (user-defined)	<p>Use this attribute for any purpose. It is not used by VCS.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: 0

Table C-3 Service group attributes (*continued*)

Service Group Attributes	Definition
UserStrGlobal (user-defined)	VCS uses this attribute in the ClusterService group. Do not modify this attribute in the ClusterService group. Use the attribute for any purpose in other service groups. <ul style="list-style-type: none">■ Type and dimension: string-scalar■ Default: 0
UserIntLocal (user-defined)	Use this attribute for any purpose. It is not used by VCS. <ul style="list-style-type: none">■ Type and dimension: integer-scalar■ Default: 0
UserStrLocal (user-defined)	Use this attribute for any purpose. It is not used by VCS. <ul style="list-style-type: none">■ Type and dimension: string-scalar■ Default: ""

System attributes

[Table C-4](#) lists the system attributes.

Table C-4 System attributes

System Attributes	Definition
AgentsStopped (system use only)	This attribute is set to 1 on a system when all agents running on the system are stopped. <ul style="list-style-type: none">■ Type and dimension: integer-scalar■ Default: Not applicable

Table C-4 System attributes (*continued*)

System Attributes	Definition
AvailableCapacity (system use only)	<p>Indicates the available capacity of the system.</p> <p>The function of this attribute depends on the value of the cluster-level attribute Statistics. If the value of the Statistics is:</p> <ul style="list-style-type: none">■ Enabled—Any change made to AvailableCapacity does not affect VCS behavior.■ MeteringOnly or Disabled—This attribute is populated based on Capacity and DynamicLoad (system attributes) and Load (service group attribute) specified using the <code>hasys -load</code> command. The key for this value is Units. <p>You cannot configure this attribute in the <code>main.cf</code> file.</p> <ul style="list-style-type: none">■ Type and dimension: float-association■ Default: Not applicable

Table C-4 System attributes (*continued*)

System Attributes	Definition
Capacity (user-defined)	<p>Represents total capacity of a system. The possible values are:</p> <ul style="list-style-type: none"> ■ Enabled—The attribute Capacity is auto-populated by meters representing system capacity in absolute units. It has all the keys specified in HostMeters, such as CPU, Mem, and Swap. The values for keys are set in corresponding units as specified in the Cluster attribute MeterUnit. You cannot configure this attribute in the main.cf file if the cluster-level attribute Statistics is set to Enabled. If the cluster-level attribute Statistics is enabled, any change made to Capacity does not affect VCS behavior. ■ MeteringOnly or Disabled—Allows you to define a value for Capacity if the value of the cluster-level attribute Statistics is set to MeteringOnly or Disabled. This value is relative to other systems in the cluster and does not reflect any real value associated with a particular system. The key for this value is Units. The default value for this attribute is 100 Units. For example, the administrator may assign a value of 200 to a 16-processor system and 100 to an 8-processor system. You can configure this attribute in the main.cf file and also modify the value at run time if Statistics is set to MeteringOnly or Disabled. ■ Type and dimension: float-association ■ Default: Depends on the value set for Statistics.
ConfigBlockCount (system use only)	<p>Number of 512-byte blocks in configuration when the system joined the cluster.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: Not applicable
ConfigChecksum (system use only)	<p>Sixteen-bit checksum of configuration identifying when the system joined the cluster.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: Not applicable

Table C-4 System attributes (*continued*)

System Attributes	Definition
ConfigDiskState (system use only)	<p>State of configuration on the disk when the system joined the cluster.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: Not applicable
ConfigFile (system use only)	<p>Directory containing the configuration files.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-scalar ■ Default: <code>/etc/VRTSvcs/conf/config</code>
ConfigInfoCnt (system use only)	<p>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.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: Not applicable
ConfigModDate (system use only)	<p>Last modification date of configuration when the system joined the cluster.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: Not applicable
CPUThresholdLevel (user-defined)	<p>Determines the threshold values for CPU utilization based on which various levels of logs are generated. The notification levels are Critical, Warning, Note, and Info, and the logs are stored in the file engine_A.log. If the Warning level is crossed, a notification is generated. The values are configurable at a system level in the cluster.</p> <ul style="list-style-type: none"> ■ For example, the administrator may set the value of CPUThresholdLevel as follows: ■ CPUThresholdLevel={Critical=95, Warning=80, Note=75, Info=60} ■ Type and dimension: integer-association ■ Default: Critical=90, Warning=80, Note=70, Info=60
CPUUsage (system use only)	<p>This attribute is deprecated. VCS monitors system resources on startup.</p>
CPUUsageMonitoring	<p>This attribute is deprecated. VCS monitors system resources on startup.</p>

Table C-4 System attributes (*continued*)

System Attributes	Definition
CurrentLimits (system use only)	<p>System-maintained calculation of current value of Limits.</p> <p>CurrentLimits = Limits - (additive value of all service group Prerequisites).</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-association ■ Default: Not applicable
DiskHbStatus (system use only)	<p>Deprecated attribute. Indicates status of communication disks on any system.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-association ■ Default: Not applicable
DynamicLoad (user-defined)	<p>System-maintained value of current dynamic load. The value is set external to VCS with the <code>hasys -load</code> command. When you specify the dynamic system load, VCS does not use the static group load.</p> <ul style="list-style-type: none"> ■ Type and dimension: float-association ■ Default: "" (none)
EngineRestarted (system use only)	<p>Indicates whether the VCS engine (HAD) was restarted by the hashadow process on a node in the cluster. The value 1 indicates that the engine was restarted; 0 indicates it was not restarted.</p> <ul style="list-style-type: none"> ■ Type and dimension: boolean-scalar ■ Default: 0
EngineVersion (system use only)	<p>Specifies the major, minor, maintenance-patch, and point-patch version of VCS.</p> <p>The value of EngineVersion attribute is in hexa-decimal format. To retrieve version information:</p> <pre>Major Version: EngineVersion >> 24 & 0xff Minor Version: EngineVersion >> 16 & 0xff Maint Patch: EngineVersion >> 8 & 0xff Point Patch: EngineVersion & 0xff</pre> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: Not applicable

Table C-4 System attributes (*continued*)

System Attributes	Definition
Frozen (system use only)	<p>Indicates if service groups can be brought online on the system. Groups cannot be brought online if the attribute value is 1.</p> <ul style="list-style-type: none">■ Type and dimension: boolean-scalar■ Default: 0
GUIIPAddr (user-defined)	<p>Determines the local IP address that VCS uses to accept connections. Incoming connections over other IP addresses are dropped. If GUIIPAddr is not set, the default behavior is to accept external connections over all configured local IP addresses.</p> <p>See “User privileges for CLI commands” on page 68.</p> <ul style="list-style-type: none">■ Type and dimension: string-scalar■ Default: ""
HostAvailableForecast (system use only)	<p>Indicates the forecasted available capacities of the systems in a cluster based on the past metered AvailableCapacity.</p> <p>The HostMonitor agent auto-populates values for this attribute, if the cluster attribute Statistics is set to Enabled. It has all the keys specified in HostMeters, such as CPU, Mem, and Swap. The values for keys are set in corresponding units as specified in the Cluster attribute MeterUnit.</p> <p>You cannot configure this attribute in main.cf.</p> <ul style="list-style-type: none">■ Type and dimension: float-association■ Default: Not applicable

Table C-4 System attributes (*continued*)

System Attributes	Definition
LicenseType (system use only)	<p>Indicates the license type of the base VCS key used by the system. Possible values are:</p> <p>0—DEMO</p> <p>1—PERMANENT</p> <p>2—PERMANENT_NODE_LOCK</p> <p>3—DEMO_NODE_LOCK</p> <p>4—NFR</p> <p>5—DEMO_EXTENSION</p> <p>6—NFR_NODE_LOCK</p> <p>7—DEMO_EXTENSION_NODE_LOCK</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: Not applicable
Limits (user-defined)	<p>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.</p> <p>The format for Limits is: Limits = { Name=Value, Name2=Value2}.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-association ■ Default: ""
LinkHbStatus (system use only)	<p>Indicates status of private network links on any system.</p> <p>Possible values include the following:</p> <p><code>LinkHbStatus = { nic1 = UP, nic2 = DOWN }</code></p> <p>Where the value UP for <i>nic1</i> means there is at least one peer in the cluster that is visible on <i>nic1</i>.</p> <p>Where the value DOWN for <i>nic2</i> means no peer in the cluster is visible on <i>nic2</i>.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-association ■ Default: Not applicable

Table C-4 System attributes (*continued*)

System Attributes	Definition
LLTNodeId (system use only)	<p>Displays the node ID defined in the file.</p> <p>%VCS_HOME%\comms\llt\llttab.txt</p> <ul style="list-style-type: none">■ Type and dimension: integer-scalar■ Default: Not applicable
LoadTimeCounter (system use only)	<p>System-maintained internal counter of how many seconds the system load has been above LoadWarningLevel. This value resets to zero anytime system load drops below the value in LoadWarningLevel.</p> <p>If the cluster-level attribute Statistics is enabled, any change made to LoadTimeCounter does not affect VCS behavior.</p> <ul style="list-style-type: none">■ Type and dimension: integer-scalar■ Default: Not applicable
LoadTimeThreshold (user-defined)	<p>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.</p> <p>If the cluster-level attribute Statistics is enabled, any change made to LoadTimeThreshold does not affect VCS behavior.</p> <ul style="list-style-type: none">■ Type and dimension: integer-scalar■ Default: 600
LoadWarningLevel (user-defined)	<p>A percentage of total capacity where load has reached a critical limit. If set to 0 overload calculations are disabled.</p> <p>For example, setting LoadWarningLevel = 80 sets the warning level to 80 percent.</p> <p>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.</p> <p>If the cluster-level attribute Statistics is enabled, any change made to LoadWarningLevel does not affect VCS behavior.</p> <ul style="list-style-type: none">■ Type and dimension: integer-scalar■ Default: 80

Table C-4 System attributes (*continued*)

System Attributes	Definition
MemThresholdLevel (user-defined)	<p>Determines the threshold values for memory utilization based on which various levels of logs are generated. The notification levels are Critical, Warning, Note, and Info, and the logs are stored in the file engine_A.log. If the Warning level is crossed, a notification is generated. The values are configurable at a system level in the cluster.</p> <p>For example, the administrator may set the value of MemThresholdLevel as follows:</p> <ul style="list-style-type: none">■ MemThresholdLevel={Critical=95, Warning=80, Note=75, Info=60}■ Type and dimension: integer-association■ Default: Critical=90, Warning=80, Note=70, Info=60

Table C-4 System attributes (*continued*)

System Attributes	Definition
MeterRecord (system use only)	<p>Acts as an internal system attribute with predefined keys. This attribute is updated only when the Cluster attribute AdpativePolicy is set to Enabled.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-association ■ Default: Not applicable <p>Possible keys are:</p> <ul style="list-style-type: none"> ■ AvailableGC: Stores the value of the cluster attribute GlobalCounter when the HostMonitor agent updates the system attribute AvailableCapacity. If the value of AvailableGC for a system in running state is older than the last 24 values of the cluster attribute GlobalCounter it indicates: <ul style="list-style-type: none"> ■ Values are not updated at the required frequency by the HostMonitor agent. ■ Values of system attributes AvailableCapacity and HostUtilization are stale. ■ ForecastGC: Stores cluster attribute GlobalCounter value when system HostAvailableForecast attribute is updated. If the value of ForecastGC for a system in running state is older than the last 72 values of the cluster attribute GlobalCounter it indicates : <ul style="list-style-type: none"> ■ HostMonitor agent is not forecasting the available capacity at the required frequency. ■ The values of the system attribute HostAvailableForecast are stale. ■ If any of the running systems in SystemList have stale value in HostAvailableForecast when FailOverPolicy is set to BiggestAvailable, then VCS does not apply BiggestAvailable policy. Instead, it considers Priority as the FailOverPolicy.
NoAutoDisable (system use only)	<p>When set to 0, this attribute autodisables service groups when the VCS engine is taken down. Groups remain autodisabled until the engine is brought up (regular membership).</p> <p>This attribute's value is updated whenever a node joins (gets into RUNNING state) or leaves the cluster. This attribute cannot be set manually.</p> <ul style="list-style-type: none"> ■ Type and dimension: boolean-scalar ■ Default: 0

Table C-4 System attributes (*continued*)

System Attributes	Definition
NodeId (system use only)	<p>System (node) identification specified in:</p> <p>%VCS_HOME%\comms\l\t\ttab.txt</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: Not applicable
OnGrpCnt (system use only)	<p>Number of groups that are online, or about to go online, on a system.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: Not applicable
ReservedCapacity (system use only)	<p>Indicates the reserved capacity on the systems for service groups which are coming online and with FailOverPolicy is set to BiggestAvailable. It has all of the keys specified in HostMeters, such as CPU, Mem, and Swap. The values for keys are set in corresponding units as specified in the Cluster attribute MeterUnit.</p> <ul style="list-style-type: none"> ■ Type and dimension: float-association ■ Default: Not applicable <p>When the service group completes online transition and after the next forecast cycle, ReservedCapacity is updated.</p> <p>You cannot configure this attribute in main.cf.</p>
ShutdownTimeout (user-defined)	<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. Veritas recommends that you set this attribute depending on the average time it takes to shut down the system.</p> <p>If you do not want to treat the system reboot as a fault, set the value for this attribute to 0.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: 120 seconds

Table C-4 System attributes (*continued*)

System Attributes	Definition
SourceFile (user-defined)	<p>File from which the configuration is read. Do not configure this attribute in main.cf.</p> <p>Make sure the path exists on all nodes before running a command that configures this attribute.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-scalar ■ Default: .\main.cf
SwapThresholdLevel (user-defined)	<p>Determines the threshold values for swap space utilization based on which various levels of logs are generated. The notification levels are Critical, Warning, Note, and Info, and the logs are stored in the file engine_A.log. If the Warning level is crossed, a notification is generated. The values are configurable at a system level in the cluster.</p> <ul style="list-style-type: none"> ■ For example, the administrator may set the value of SwapThresholdLevel as follows: ■ SwapThresholdLevel={Critical=95, Warning=80, Note=75, Info=60} ■ Type and dimension: integer-association ■ Default: Critical=90, Warning=80, Note=70, Info=60
SysInfo (system use only)	<p>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.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-scalar ■ Default: Not applicable
SysName (system use only)	<p>Indicates the system name.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-scalar ■ Default: Not applicable
SysState (system use only)	<p>Indicates system states, such as RUNNING, FAULTED, EXITED, etc.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: Not applicable
SystemLocation (user-defined)	<p>Indicates the location of the system.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-scalar ■ Default: ""

Table C-4 System attributes (*continued*)

System Attributes	Definition
SystemOwner (user-defined)	<p>Use this attribute 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. Note that while VCS logs most events, not all events trigger notifications.</p> <p>Make sure to set the severity level at which you want notifications to SystemOwner or to at least one recipient defined in the Smtprcipients attribute of the NotifierMngr agent.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-scalar ■ Default: "" ■ Example: "unknown"
SystemRecipients (user-defined)	<p>This attribute is used for VCS email notification. VCS sends email notification to persons designated in this attribute when events related to the system occur and when the event's severity level is equal to or greater than the level specified in the attribute.</p> <p>Make sure to set the severity level at which you want notifications to be sent to SystemRecipients or to at least one recipient defined in the Smtprcipients attribute of the NotifierMngr agent.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-association ■ email id: The e-mail address of the person registered as a recipient for notification. severity: The minimum level of severity at which notifications must be sent.
TFrozen (user-defined)	<p>Indicates whether a service group can be brought online on a node. Service group cannot be brought online if the value of this attribute is 1.</p> <ul style="list-style-type: none"> ■ Type and dimension: boolean-scalar ■ Default: 0
TRSE (system use only)	<p>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.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: Not applicable

Table C-4 System attributes (*continued*)

System Attributes	Definition
UpDownState (system use only)	<p>This attribute has four values:</p> <p>Down (0): System is powered off, or GAB and LLT are not running on the system.</p> <p>Up but not in cluster membership (1): GAB and LLT are running but the VCS engine is not.</p> <p>Up and in jeopardy (2): The system is up and part of cluster membership, but only one network link (LLT) remains.</p> <p>Up (3): The system is up and part of cluster membership, and has at least two links to the cluster.</p> <ul style="list-style-type: none">■ Type and dimension: integer-scalar■ Default: Not applicable
UserInt (user-defined)	<p>Stores integer values you want to use. VCS does not interpret the value of this attribute.</p> <ul style="list-style-type: none">■ Type and dimension: integer-scalar■ Default: 0
VCSFeatures (system use only)	<p>Indicates which VCS features are enabled. Possible values are:</p> <p>0—No features enabled (VCS Simulator)</p> <p>1—L3+ is enabled</p> <p>2—Global Cluster Option is enabled</p> <p>Even though VCSFeatures attribute is an integer attribute, when you query the value with the <code>hasys -value</code> command or the <code>hasys -display</code> command, it displays as the string L10N for value 1 and DR for value 2.</p> <ul style="list-style-type: none">■ Type and dimension: integer-scalar■ Default: Not applicable

Cluster attributes

Table C-5 lists the cluster attributes.

Table C-5 Cluster attributes

Cluster Attributes	Definition
AdministratorGroups (user-defined)	<p>List of operating system user account groups that have administrative privileges on the cluster. This attribute applies to clusters running in secure mode.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-keylist ■ Default: ""
Administrators (user-defined)	<p>Contains list of users with Administrator privileges.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-keylist ■ Default: ""
AuthorizationControl (user-defined)	<p>This attribute applies to clusters running in secure mode. It defines how VCS assigns cluster privileges to operating system (OS) users that have local or domain administrator privileges. The OS users must be defined as VCS users before modifying the attribute's default value.</p> <p>The attribute can take the following values:</p> <ul style="list-style-type: none"> ■ DEFAULT: Assigns cluster administrator privileges to users with local administrator and domain administrator privileges. ■ NONE: Does not assign cluster administrator privileges to users with local administrator and domain administrator privileges. Windows services running under local system accounts get cluster administrator privileges when they connect to VCS. You can override this setting by defining privileges for local and domain administrators in the VCS user list. ■ LOCAL : Assigns cluster administrator privileges to local administrators, but not to domain administrators. You can override this setting by defining privileges for domain administrators in the VCS user list. ■ GLOBAL : Assigns cluster administrator privileges to domain administrators, but not to local administrators. You can override this setting by defining privileges for local administrators in the VCS user list.
AutoClearQ (System use only)	<p>Lists the service groups scheduled to be auto-cleared. It also indicates the time at which the auto-clear for the group will be performed.</p>
AutoStartTimeout (user-defined)	<p>If the local cluster cannot communicate with one or more remote clusters, this attribute specifies the number of seconds the VCS engine waits before initiating the AutoStart process for an AutoStart global service group.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: 150 seconds

Table C-5 Cluster attributes (*continued*)

Cluster Attributes	Definition
AutoAddSystemtoCSG (user-defined)	<p>Indicates whether the newly joined or added systems in cluster become part of the SystemList of the ClusterService service group if the service group is configured. The value 1 (default) indicates that the new systems are added to SystemList of ClusterService. The value 0 indicates that the new systems are not added to SystemList of ClusterService.</p> <ul style="list-style-type: none">■ Type and dimension: integer-scalar■ Default: 1
BackupInterval (user-defined)	<p>Time period in minutes after which VCS backs up the configuration files if the configuration is in read-write mode.</p> <p>The value 0 indicates VCS does not back up configuration files. Set this attribute to at least 3.</p> <p>See “Scheduling automatic backups for VCS configuration files” on page 106.</p> <ul style="list-style-type: none">■ Type and dimension: integer-scalar■ Default: 0
ClusState (system use only)	<p>Indicates the current state of the cluster.</p> <ul style="list-style-type: none">■ Type and dimension: integer-scalar■ Default: Not applicable.
ClusterAddress (user-defined)	<p>Specifies the cluster’s virtual IP address (used by a remote cluster when connecting to the local cluster).</p> <ul style="list-style-type: none">■ Type and dimension: string-scalar■ Default: ""
ClusterLocation (user-defined)	<p>Specifies the location of the cluster.</p> <ul style="list-style-type: none">■ Type and dimension: string-scalar■ Default: ""
ClusterName (user-defined)	<p>The name of cluster.</p> <ul style="list-style-type: none">■ Type and dimension: string-scalar■ Default: ""

Table C-5 Cluster attributes (*continued*)

Cluster Attributes	Definition
ClusterOwner (user-defined)	<p>This attribute used for VCS notification. VCS sends notifications to persons designated in this attribute when an event occurs related to the cluster. Note that while VCS logs most events, not all events trigger notifications.</p> <p>Make sure to set the severity level at which you want notifications to be sent to ClusterOwner or to at least one recipient defined in the Smtprcipients attribute of the NotifierMngr agent.</p> <p>See “About VCS event notification” on page 357.</p> <ul style="list-style-type: none">■ Type and dimension: string-scalar■ Default: ""■ Example: "jdoe@example.com"
ClusterRecipients (user-defined)	<p>This attribute is used for VCS email notification. VCS sends email notification to persons designated in this attribute when events related to the cluster occur and when the event's severity level is equal to or greater than the level specified in the attribute.</p> <p>Make sure to set the severity level at which you want notifications to be sent to ClusterRecipients or to at least one recipient defined in the Smtprcipients attribute of the NotifierMngr agent.</p> <ul style="list-style-type: none">■ Type and dimension: string-association■ email id: The e-mail address of the person registered as a recipient for notification.■ severity: The minimum level of severity at which notifications must be sent.
ClusterTime (system use only)	<p>The number of seconds since January 1, 1970. This is defined by the lowest node in running state.</p> <ul style="list-style-type: none">■ Type and dimension: string-scalar■ Default: Not applicable
CompareRSM (system use only)	<p>Indicates if VCS engine is to verify that replicated state machine is consistent. This can be set by running the hadebug command.</p> <ul style="list-style-type: none">■ Type and dimension: integer-scalar■ Default: 0
ConnectorState (system use only)	<p>Indicates the state of the wide-area connector (wac). If 0, wac is not running. If 1, wac is running and communicating with the VCS engine.</p> <ul style="list-style-type: none">■ Type and dimension: integer-scalar■ Default: Not applicable.

Table C-5 Cluster attributes (*continued*)

Cluster Attributes	Definition
CounterInterval (user-defined)	<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> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: 5
CounterMissAction (user-defined)	<p>Specifies the action that must be performed when the GlobalCounter is not updated for CounterMissTolerance times the CounterInterval. Possible values are LogOnly and Trigger. If you set CounterMissAction to LogOnly, the system logs the message in Engine Log and Syslog. If you set CounterMissAction to Trigger, the system invokes a trigger which has default action of collecting the comms tar file.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-scalar ■ Default: LogOnly
CounterMissTolerance (user-defined)	<p>Specifies the time interval that can lapse since the last update of GlobalCounter before VCS reports an issue. If the GlobalCounter does not update within CounterMissTolerance times CounterInterval, VCS reports the issue. Depending on the CounterMissAction.value, appropriate action is performed.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: 20
CredRenewFrequency (user-defined)	<p>The number of days after which the VCS engine renews its credentials with the authentication broker. For example, the value 5 indicates that credentials are renewed every 5 days; the value 0 indicates that credentials are not renewed.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default = 0
DumpingMembership (system use only)	<p>Indicates that the engine is writing or dumping the configuration to disk.</p> <ul style="list-style-type: none"> ■ Type and dimension: vector ■ Default: Not applicable.
EnableFFDC (user-defined)	<p>Enables or disables FFDC logging. By default, FFDC logging is enabled.</p> <ul style="list-style-type: none"> ■ Type and dimension: boolean-scalar ■ Default: 1

Table C-5 Cluster attributes (*continued*)

Cluster Attributes	Definition
EnableVMAutoDiscovery (user-defined)	<p>Enables or disables auto discovery of virtual machines. By default, auto discovery of virtual machines is disabled.</p> <ul style="list-style-type: none">■ Type and dimension: integer-scalar■ Default: 0
EnablePBF (user-defined)	<p>Enables or disables priority based failover. When set to 1 (one), VCS gives priority to the online of high priority service group, by ensuring that its Load requirement is met on the system.</p> <ul style="list-style-type: none">■ Type and dimension: boolean-scalar■ Default: 0 (zero)
EnginePriority (user-defined)	<p>The priority in which HAD runs. Generally, a greater priority value indicates higher scheduling priority. A range of priority values is assigned to each scheduling class. For more information on the range of priority values, see the operating system documentation.</p> <ul style="list-style-type: none">■ Type and dimension: string-scalar■ Default: ""
EngineShutdown (user-defined)	<p>Defines the options for the hastop command. The attribute can assume the following values:</p> <p>Enable—Process all hastop commands. This is the default behavior.</p> <p>Disable—Reject all hastop commands.</p> <p>DisableClusStop—Do not process the hastop -all command; process all other hastop commands.</p> <p>PromptClusStop—Prompt for user confirmation before running the hastop -all command; process all other hastop commands.</p> <p>PromptLocal—Prompt for user confirmation before running the hastop -local command; reject all other hastop commands.</p> <p>PromptAlways—Prompt for user confirmation before running any hastop command.</p> <ul style="list-style-type: none">■ Type and dimension: string-scalar■ Default: Enable

Table C-5 Cluster attributes (*continued*)

Cluster Attributes	Definition
FipsMode (system use only)	<p>Indicates whether FIPS mode is enabled for the cluster. The value depends on the mode of the broker on the system. If FipsMode is set to 1, FIPS mode is enabled. If FipsMode is set to 0, FIPS mode is disabled.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer -scalar ■ Default: Not applicable <p>You can verify the value of FipsMode as follows:</p> <pre># haclus -value FipsMode</pre>
GlobalCounter (system use only)	<p>This counter increases incrementally by one for each counter interval. It increases when the broadcast is received.</p> <p>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 600-second default value of the ShutdownTimeout attribute, means if system goes down within 120 increments of GlobalCounter, it is treated as a fault. Change the value of the CounterInterval attribute to modify the default value of GlobalCounter increment.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: Not applicable
Guests (user-defined)	<p>List of operating system user accounts that have Guest privileges on the cluster.</p> <p>This attribute is valid clusters running in secure mode.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-keylist ■ Default: ""
GuestGroups (user-defined)	<p>List of operating system user groups that have Guest privilege on the cluster.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-keylist ■ Default: Not applicable
DefaultGuestAccess (user-defined)	<p>Indicates whether any authenticated user should have guest access to the cluster by default. The default guest access can be:</p> <ul style="list-style-type: none"> ■ 0: Guest access for privileged users only. ■ 1: Guest access for everyone. ■ Type and dimension: boolean-scalar ■ Default: 0

Table C-5 Cluster attributes (*continued*)

Cluster Attributes	Definition
GroupLimit (user-defined)	<p>Maximum number of service groups.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: 200
HacliUserLevel (user-defined)	<p>This attribute has two, case-sensitive values:</p> <p>NONE—hacli is disabled for all users regardless of role.</p> <p>COMMANDROOT—hacli is enabled for root only.</p> <p>Note: The command <code>haclus -modify HacliUserLevel</code> can be executed by root only.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-scalar ■ Default: NONE
HostAvailableMeters (System use only)	<p>Lists the meters that are available for measuring system resources. You cannot configure this attribute in the main.cf file.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-association <p>Keys are the names of parameters and values are the names of meter libraries.</p>
HostMeters (user-defined)	<ul style="list-style-type: none"> ■ Type and dimension: string-keylist ■ You can configure this attribute in the main.cf file. You cannot modify the value at run time.
LockMemory (user-defined)	<p>Controls the locking of VCS engine pages in memory. This attribute has the following values. Values are case-sensitive:</p> <p>ALL: Locks all current and future pages.</p> <p>CURRENT: Locks current pages.</p> <p>NONE: Does not lock any pages.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-scalar ■ Default: ALL
LogClusterUUID (user-defined)	<p>Enables or disables logging of the cluster UUID in each log message. By default, cluster UUID is not logged.</p> <ul style="list-style-type: none"> ■ Type and dimension: boolean-scalar ■ Default: 0

Table C-5 Cluster attributes (*continued*)

Cluster Attributes	Definition
LogSize (user-defined)	<p>Indicates the size of engine log files in bytes.</p> <p>Minimum value is = 65536 (equal to 64KB)</p> <p>Maximum value = 134217728 (equal to 128MB)</p> <ul style="list-style-type: none">■ Type and dimension: integer-scalar■ Default: 134217728 (128 MB)
MeterControl (user-defined)	<p>Indicates the intervals at which metering and forecasting for the system attribute AvailableCapacity are done for the keys specified in HostMeters.</p> <ul style="list-style-type: none">■ Type and dimension: integer-association This attribute includes the following keys:■ MeterInterval Frequency in seconds at which metering is done by the HostMonitor agent. The value for this key can equal or exceed 30. The default value is 120 indicating that the HostMonitor agent meters available capacity and updates the System attribute AvailableCapacity every 120 seconds. The HostMonitor agent checks for changes in the available capacity for every monitoring cycle and when there is a change, the HostMonitor agent updates the values in the same monitoring cycle . The MeterInterval value applies only if Statistics is set to Enabled or MeterHostOnly.■ ForecastCycle The number of metering cycles after which forecasting of available capacity is done. The value for this key can equal or exceed 1. The default value is 3 indicating that forecasting of available capacity is done after every 3 metering cycles. Assuming the default MeterInterval value of 120 seconds, forecasting is done after 360 seconds or 6 minutes. The ForecastCycle value applies only if Statistics is set to Enabled. <p>You can configure this attribute in main.cf. You cannot modify the value at run time. The values of MeterInterval and ForecastCycle apply to all keys of HostMeters.</p>

Table C-5 Cluster attributes (*continued*)

Cluster Attributes	Definition
MeterUnit	<p>Represents units for parameters that are metered.</p> <ul style="list-style-type: none"> Type and dimension: string-association <p>You can configure this attribute in main.cf; if configured in main.cf, then it must contain units for all the keys as specified in HostMeters. You cannot modify the value at run time.</p> <ul style="list-style-type: none"> AvailableCapacity HostAvailableForecast Capacity ReservedCapacity <p>The values of keys such as Mem and Swap can be represented in MB or GB, and CPU can be represented in CPU, MHz or GHz.</p>
MeterWeight (user-defined)	<p>Indicates the default meter weight for the service groups in the cluster. You can configure this attribute in the main.cf file, but you cannot modify the value at run time. If the attribute is defined in the main.cf file, it must have at least one key defined. The weight for the key must be in the range of 0 to 10. Only keys from HostAvailableMeters are allowed in this attribute.</p> <ul style="list-style-type: none"> Type and dimension: integer-association
Notifier (system use only)	<p>Indicates the status of the notifier in the cluster; specifically:</p> <p>State—Current state of notifier, such as whether or not it is connected to VCS.</p> <p>Host—The host on which notifier is currently running or was last running. Default = None</p> <p>Severity—The severity level of messages queued by VCS for notifier. Values include Information, Warning, Error, and SevereError. Default = Warning</p> <p>Queue—The size of queue for messages queued by VCS for notifier.</p> <ul style="list-style-type: none"> Type and dimension: string-association Default: Different values for each parameter.
OpenExternalCommunicationPort (user-defined)	<p>Indicates whether communication over the external communication port for VCS is allowed or not. By default, the external communication port for VCS is 14141.</p> <ul style="list-style-type: none"> Type and dimension: string-scalar Valid values: YES, NO Default: YES YES: The external communication port for VCS is open. NO: The external communication port for VCS is not open.

Table C-5 Cluster attributes (*continued*)

Cluster Attributes	Definition
OperatorGroups (user-defined)	<p>List of operating system user groups that have Operator privileges on the cluster.</p> <p>This attribute is valid clusters running in secure mode.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-keylist ■ Default: ""
Operators (user-defined)	<p>List of users with Cluster Operator privileges.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-keylist ■ Default: ""
PanicOnNoMem (user-defined)	<p>Indicate the action that you want VCS engine (HAD) to take if it cannot receive messages from GAB due to low-memory.</p> <ul style="list-style-type: none"> ■ If the value is 0, VCS exits with warnings. ■ If the value is 1, VCS calls the GAB library routine to panic the system. ■ Default: 0
PrintMsg (user-defined)	<p>Enables logging TagM messages in engine log if set to 1.</p> <ul style="list-style-type: none"> ■ Type and dimension: boolean-scalar ■ Default: 0
ProcessPriority (user-defined)	<p>The priority of processes created by the VCS engine. For example triggers.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-scalar ■ Default: ""
ReadOnly (user-defined)	<p>Indicates that cluster is in read-only mode.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: 1
ResourceLimit (user-defined)	<p>Maximum number of resources.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: 5000
SecInfo256 (user-defined)	<p>Enables creation of secure passwords when this attribute is added to the <code>main.cf</code> file with the security key as the value of the attribute.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-scalar ■ Default: "" <p>See "Encrypting agent passwords" on page 99.</p>

Table C-5 Cluster attributes (*continued*)

Cluster Attributes	Definition
IV256 (user-defined)	<p>Enables creation of secure passwords when this attribute is added to the <code>main.cf</code> file. This initialization vector is a fixed-size input to a cryptographic primitive that is typically required to be random. It adds randomness to the beginning of the encryption process.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-scalar ■ Default: "" <p>See “Encrypting agent passwords” on page 99.</p>
SecInfoLevel (user-defined)	<p>Denotes the password encryption privilege level.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-scalar ■ Default: R <p>See “Encrypting agent passwords” on page 99.</p>
SecureClus (user-defined)	<p>Indicates whether the cluster runs in secure mode. The value 1 indicates the cluster runs in secure mode. This attribute cannot be modified when VCS is running.</p> <ul style="list-style-type: none"> ■ Type and dimension: boolean-scalar ■ Default: 0
SiteAware (user-defined)	<p>Indicates whether sites are configured for a cluster or not.</p> <ul style="list-style-type: none"> ■ Type and dimension: boolean-scalar ■ Default: 0 <p>Possible values are:</p> <ul style="list-style-type: none"> ■ 1: Sites are configured. ■ 0: Sites are not configured. <p>You can configure a site from Veritas InfoScale Operations Manager . This attribute will be automatically set to 1 when configured using Veritas InfoScale Operations Manager. If site information is not configured for some nodes in the cluster, those nodes are placed under a default site that has the lowest preference.</p>
SourceFile (user-defined)	<p>File from which the configuration is read. Do not configure this attribute in <code>main.cf</code>. Make sure the path exists on all nodes before running a command that configures this attribute.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-scalar ■ Default: Not applicable.

Table C-5 Cluster attributes (*continued*)

Cluster Attributes	Definition
Statistics (user-defined)	<p>Indicates if statistics gathering is enabled and whether the FailOverPolicy can be set to BiggestAvailable.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-scalar ■ Default: Enabled <p>You cannot modify the value at run time.</p> <p>Possible values are:</p> <ul style="list-style-type: none"> ■ Enabled: The HostMonitor agent meters host utilization and forecasts the available capacity for the systems in the cluster. With this value set, FailOverPolicy for any service group cannot be set to Load. ■ MeterHostOnly: The HostMonitor agent meters host utilization but it does not forecast the available capacity for the systems in the cluster. The service group attribute FailOverPolicy cannot be set to BiggestAvailable. ■ Disabled: The HostMonitor agent is not started. Both metering of host utilization and forecasting of available capacity are disabled. The service group attribute FailOverPolicy cannot be set to BiggestAvailable. <p>See "Service group attributes" on page 546.</p>
Stewards (user-defined)	<p>The IP address and hostname of systems running the steward process.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-keylist ■ {}
SystemRebootAction (user-defined)	<p>Determines whether frozen service groups are ignored on system reboot.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-keylist ■ Default: "" <p>If the SystemRebootAction value is IgnoreFrozenGroup, VCS ignores service groups that are frozen (TFrozen and Frozen) and takes the remaining service groups offline. If the frozen service groups have firm dependencies or hard dependencies on any other service groups which are not frozen, VCS gives an error.</p> <p>If the SystemRebootAction value is "", VCS tries to take all service groups offline. Because VCS cannot be gracefully stopped on a node where a frozen service group is online, applications on the node might get killed.</p> <p>Note: The SystemRebootAction attribute applies only on system reboot and system shutdown.</p>
TypeLimit (user-defined)	<p>Maximum number of resource types.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: 100

Table C-5 Cluster attributes (*continued*)

Cluster Attributes	Definition
UserNames (user-defined)	List of VCS users. The installer uses <code>admin</code> as the default user name. <ul style="list-style-type: none">■ Type and dimension: string-association■ Default: ""
VCSFeatures (system use only)	Indicates which VCS features are enabled. Possible values are: 0—No features are enabled (VCS Simulator) 1—L3+ is enabled 2—Global Cluster Option is enabled Even though the VCSFeatures is an integer attribute, when you query the value with the <code>haclus -value</code> command or the <code>haclus -display</code> command, it displays as the string L10N for value 1 and DR for value 2. <ul style="list-style-type: none">■ Type and dimension: integer-scalar■ Default: Not applicable.
VCSType (system use only)	Denotes the mode for which VCS is licensed. Even though the VCSType is an integer attribute, when you query the value with the <code>haclus -value</code> command or the <code>haclus -display</code> command, it displays as the string UNKNOWN_MODE for value 0 and VCS for value 7. <ul style="list-style-type: none">■ Type and dimension: integer-scalar■ Default: Not applicable
WACPort (user-defined)	The TCP port on which the <code>wac</code> (Wide-Area Connector) process on the local cluster listens for connection from remote clusters. Type and dimension: integer-scalar <ul style="list-style-type: none">■ Default: 14155

Heartbeat attributes (for global clusters)

Table C-6 lists the heartbeat attributes. These attributes apply to global clusters.

Table C-6 Heartbeat attributes

Heartbeat Attributes	Definition
AgentState (system use only)	The state of the heartbeat agent. <ul style="list-style-type: none">■ Type and dimension: integer-scalar■ Default: INIT

Table C-6 Heartbeat attributes (*continued*)

Heartbeat Attributes	Definition
Arguments (user-defined)	<p>List of arguments to be passed to the agent functions. For the lcmp agent, this attribute can be the IP address of the remote cluster.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-vector ■ Default: ""
AYAIInterval (user-defined)	<p>The interval in seconds between two heartbeats.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: 60 seconds
AYARetryLimit (user-defined)	<p>The maximum number of lost heartbeats before the agent reports that heartbeat to the cluster is down.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: 3
AYATimeout (user-defined)	<p>The maximum time (in seconds) that the agent will wait for a heartbeat AYA function to return ALIVE or DOWN before being canceled.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: 30
CleanTimeOut (user-defined)	<p>Number of seconds within which the Clean function must complete or be canceled.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: 300 seconds
ClusterList (user-defined)	<p>List of remote clusters.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-keylist ■ Default: ""
InitTimeout (user-defined)	<p>Number of seconds within which the Initialize function must complete or be canceled.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: 300 seconds
LogDbg (user-defined)	<p>The log level for the heartbeat.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-keylist ■ Default: ""

Table C-6 Heartbeat attributes (*continued*)

Heartbeat Attributes	Definition
State	The state of the heartbeat. <ul style="list-style-type: none">■ Type and dimension: integer-scalar■ Default: Not applicable
StartTimeout (user-defined)	Number of seconds within which the Start function must complete or be canceled. <ul style="list-style-type: none">■ Type and dimension: integer-scalar■ Default: 300 seconds
StopTimeout (user-defined)	Number of seconds within which the Stop function must complete or be canceled without stopping the heartbeat. <ul style="list-style-type: none">■ Type and dimension: integer-scalar■ Default: 300 seconds

Remote cluster attributes

[Table C-7](#) lists the RemoteCluster attributes. These attributes apply to remote clusters.

Table C-7 Remote cluster attributes

Remote cluster Attributes	Definition
AdministratorGroups (system use only)	List of operating system user account groups that have administrative privileges on the cluster. This attribute applies to clusters running in secure mode. <ul style="list-style-type: none">■ Type and dimension: string-keylist■ Default: " "
Administrators (system use only)	Contains list of users with Administrator privileges. <ul style="list-style-type: none">■ Type and dimension: string-keylist■ Default: ""
CID (system use only)	The CID of the remote cluster. See “Cluster attributes” on page 583.

Table C-7 Remote cluster attributes (*continued*)

Remote cluster Attributes	Definition
ClusState (system use only)	Indicates the current state of the remote cluster as perceived by the local cluster. <ul style="list-style-type: none">■ Type and dimension: integer-scalar■ Default: Not applicable
ClusterAddress (user-defined)	Specifies the remote cluster's virtual IP address, which is used to connect to the remote cluster by the local cluster. <ul style="list-style-type: none">■ Type and dimension: string-scalar■ Default: ""
ClusterName (system use only)	The name of cluster. <ul style="list-style-type: none">■ Type and dimension: string-scalar■ Default: ""
ConnectTimeout (user-defined)	Specifies the time in milliseconds for establishing the WAC to WAC connection. <ul style="list-style-type: none">■ Type and dimension: integer-scalar■ Default: 300
DeclaredState (user-defined)	Specifies the declared state of the remote cluster after its cluster state is transitioned to FAULTED. <ul style="list-style-type: none">■ Type and dimension: string-scalar■ Default: "" The value can be set to one of the following values: <ul style="list-style-type: none">■ Disaster■ Outage■ Disconnect■ Replica

Table C-7 Remote cluster attributes (*continued*)

Remote cluster Attributes	Definition
EngineVersion (system use only)	<p>Specifies the major, minor, maintenance-patch, and point-patch version of VCS.</p> <p>The value of EngineVersion attribute is in hexa-decimal format. To retrieve version information:</p> <pre>Major Version: EngineVersion >> 24 & 0xff Minor Version: EngineVersion >> 16 & 0xff Maint Patch: EngineVersion >> 8 & 0xff Point Patch: EngineVersion & 0xff</pre> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: Not applicable
Guests (system use only)	<p>List of operating system user accounts that have Guest privileges on the cluster.</p> <p>This attribute is valid for clusters running in secure mode.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-keylist ■ Default: ""
OperatorGroups (system use only)	<p>List of operating system user groups that have Operator privileges on the cluster. This attribute is valid for clusters running in secure mode.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-keylist ■ Default: 300 seconds
Operators (system use only)	<p>List of users with Cluster Operator privileges.</p> <ul style="list-style-type: none"> ■ Type and dimension: string-keylist ■ Default: ""
RemoteConnectInterval (user-defined)	<p>Specifies the time in seconds between two successive attempts to connect to the remote cluster.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: 5
SocketTimeout (user-defined)	<p>Specifies the time in seconds for WAC to WAC heartbeat. If no IAA is received in the specified time, connection with the remote WAC is assumed to be broken.</p> <ul style="list-style-type: none"> ■ Type and dimension: integer-scalar ■ Default: 180

Table C-7 Remote cluster attributes (*continued*)

Remote cluster Attributes	Definition
SourceFile (system use only)	<p>File from which the configuration is read. Do not configure this attribute in main.cf.</p> <p>Make sure the path exists on all nodes before running a command that configures this attribute.</p> <ul style="list-style-type: none">■ Type and dimension: string-scalar■ Default: Not applicable.
VCSFeatures (system use only)	<p>Indicates which VCS features are enabled. Possible values are:</p> <p>0—No features are enabled (VCS Simulator)</p> <p>1—L3+ is enabled</p> <p>2—Global Cluster Option is enabled</p> <p>Even though the VCSFeatures is an integer attribute, when you query the value with the haclus -value command or the haclus -display command, it displays as the string L10N for value 1 and DR for value 2.</p> <ul style="list-style-type: none">■ Type and dimension: integer-scalar■ Default: Not applicable.
VCSMode (system use only)	<p>Denotes the mode for which VCS is licensed.</p> <p>Even though the VCSMode is an integer attribute, when you query the value with the haclus -value command or the haclus -display command, it displays as the string UNKNOWN_MODE for value 0 and VCS for value 7.</p> <ul style="list-style-type: none">■ Type and dimension: integer-scalar■ Default: Not applicable
WACPort (system use only)	<p>The TCP port on which the wac (Wide-Area Connector) process on the remote cluster listens for connection from other clusters.</p> <ul style="list-style-type: none">■ Type and dimension: integer-scalar■ Default: 14155

Configuring LLT over UDP

This appendix includes the following topics:

- [About configuring LLT over UDP](#)
- [When to use LLT over UDP](#)
- [LLT over UDP configuration](#)
- [Sample configuration: Direct-attached links](#)
- [Sample configuration: Links crossing IP routers](#)
- [Issues and limitations](#)

About configuring LLT over UDP

VCS provides the option of using LLT over the UDP (User Datagram Protocol) layer for clusters using wide-area networks and routers. UDP makes LLT packets routable and thus able to span longer distances more economically.

The VCS Cluster Configuration Wizard (VCW) provides the necessary configuration options for using LLT over the UDP. You can configure LLT over UDP while configuring the cluster using VCW.

When to use LLT over UDP

Use LLT over UDP in the following situations:

- LLT must be used over WANs
- When hardware, such as blade servers, do not support LLT over Ethernet

Because LLT over UDP is slower than LLT over Ethernet, LLT over UDP should only be used when the hardware configuration makes it necessary.

LLT over UDP configuration

The following is a checklist for configuring LLT over UDP:

- Make sure that each NIC has an IP address configured before configuring LLT. Each link must be in a different subnet. See the examples in the following sections.
- Make sure that each link has a unique UDP port; do not assign well-known ports. See [“Selecting UDP ports”](#) on page 604.
- Set the broadcast address correctly for direct-attached (non-routed) links.
- For links that cross an IP router, disable broadcast features and specify the IP address of each link manually in the `llttab` file. The default path for the file is `%VCS_HOME%\comms\llt\llttab.txt`. See [“Sample configuration: Links crossing IP routers”](#) on page 606.

The link command in the llttab file

The following table describes the fields of the link command shown in the `llttab` file examples that follow.

See [“Sample configuration: Direct-attached links”](#) on page 605.

See [“Sample configuration: Links crossing IP routers”](#) on page 606.

Note that some of these fields differ from the command for standard LLT links.

<tag-name>	A unique string that is used as a tag by LLT; for example <code>link1</code> , <code>link2</code> , ...
<device>	The device path of the UDP protocol; for example <code>udp</code>
<node-range>	Nodes using the link. "-" indicates <i>all</i> cluster nodes are to be configured for this link.
<link-type>	Type of link; must be "udp" for LLT over UDP
<udp-port>	Unique UDP port in range of 49152-65535 for the link. See “Selecting UDP ports” on page 604.
<MTU>	"-" is the default, which has a value of 8192. The value may be increased or decreased depending on the configuration. The <code>lltstat -l</code> command can display the current value.
<IP address>	IP address of the link on the local node.

- <bcast-address>
- for clusters having broadcasts enabled, specify the value of the subnet broadcast address
 - "-" is the default for clusters spanning routers

The set-addr command in the llttab file

The `set-addr` command in the `llttab` file is required when the broadcast feature of LLT is disabled, such as when LLT must cross IP routers. The following table describes the fields of the `set-addr` command.

See [“Sample configuration: Links crossing IP routers”](#) on page 606.

<node-id>	The ID of the cluster node; for example, 0.
<link tag-name>	The string used by LLT to identify the link; for example <code>link1</code> , <code>link2</code> , ..
<address>	IP address assigned to the link on the peer node.

Selecting UDP ports

The following list provide a range of ports that you can use when selecting a UDP port:

- When selecting a UDP port, select an available 16-bit integer. Use available ports (that is, ports that are not in use)] in the private range 49152 to 65535
- Do not use:
 - Ports from the range of well-known ports, 0 to 1023
 - Ports from the range of registered ports, 1024 to 49151

To check which ports are defined as defaults for a node, examine the file `C:\WINDOWS\system32\drivers\etc>services`. You should also use the `netstat` command to list the ports currently in use. For example:

```
# netstat -a -p UDP
```

Proto	Local Address	Foreign Address	State
UDP	THORPC111:snmp	*:*	
UDP	THORPC111:snmptrap	*:*	
UDP	THORPC111:microsoft-ds	*:*	
UDP	THORPC111:isakmp	*:*	
UDP	THORPC111:1027	*:*	
UDP	THORPC111:1028	*:*	

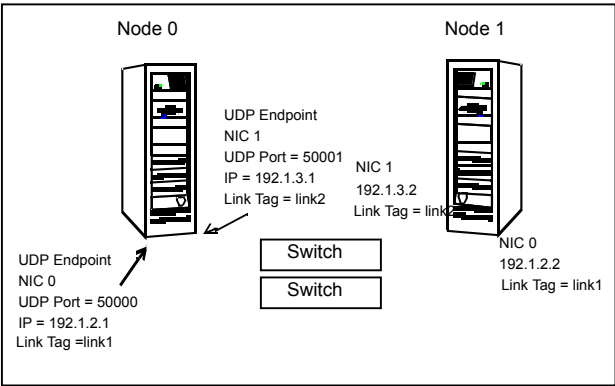
```
UDP      THORPC111:1029      *: *
UDP      THORPC111:1030      *: *
UDP      THORPC111:1059      *: *
UDP      THORPC111:1063      *: *
UDP      THORPC111:4219      *: *
UDP      THORPC111:4500      *: *
UDP      THORPC111:ntp       *: *
UDP      THORPC111:netbios-ns *: *
UDP      THORPC111:netbios-dgm *: *
UDP      THORPC111:ntp       *: *
UDP      THORPC111:1646      *: *
UDP      THORPC111:3217      *: *
UDP      THORPC111:3219      *: *
UDP      THORPC111:3456      *: *
```

Look in the UDP section of the output; UDP ports listed under `Local Address` are already in use. If a port is listed in the `services` file, its associated name is displayed rather than the port number in the output of the `netstat` command.

Sample configuration: Direct-attached links

Figure D-1 depicts a typical configuration of direct-attached links employing LLT over UDP.

Figure D-1 Direct-attached links employing LLT over UDP



The configuration represented by the following `llttab` file for Node 0 has directly attached crossover links or links connected through a hub or switch. These links do not cross routers.

Because LLT broadcasts requests to peer nodes to discover their addresses, the addresses of peer nodes do not need to be specified in the `llttab` file using the `set-addr` command. For direct attached links, you need to set the broadcast address of the links in the `llttab` file. Verify that the IP addresses and broadcast addresses are set correctly.

```
set-node Node0
set-cluster 1
#configure Links
#link <tag-name> <device> <node-range> <link-type> <udp port>
<MTU> <IP-address> <bcast-address>
link link1 udp - udp 50000 - 192.1.2.1 192.1.2.255
link link2 udp - udp 50001 - 192.1.3.1 192.1.3.255
```

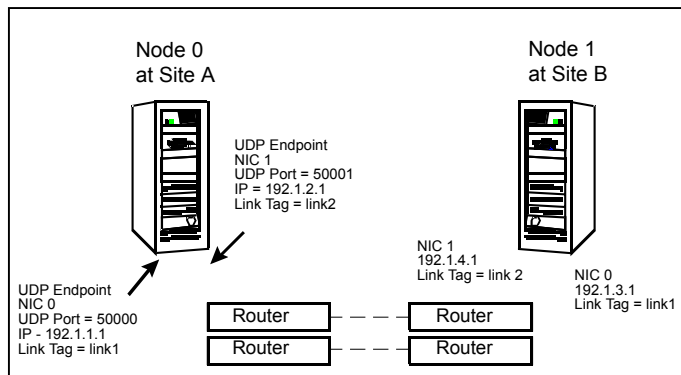
The file for Node 1 would resemble:

```
set-node Node1
set-cluster 1
#configure Links
#link <tag-name> <device> <node-range> <link-type> <udp port>
<MTU> <IP-address> <bcast-address>
link link1 udp - udp 50000 - 192.1.2.2 192.1.2.255
link link2 udp - udp 50001 - 192.1.3.2 192.1.3.255
```

Sample configuration: Links crossing IP routers

Figure D-2 depicts a typical configuration of links crossing an IP router employing LLT over UDP. The illustration shows just two nodes of a four-node cluster.

Figure D-2 Links crossing an IP router employing LLT over UDP



The configuration represented by the following `llttab` file for Node 1 has links crossing IP routers. Notice that IP addresses are shown for each link on each peer node. The broadcast features are disabled because LLT is unable to broadcast requests for addresses across routers, so the broadcast address does not need to be set in the `link` command of the `llttab` file.

```
set-node Node1
set-cluster 1

link link1 udp - udp 50000 - 192.1.3.1 -
link link2 udp - udp 50001 - 192.1.4.1 -

#set address of each link for all peer nodes in the cluster
#format: set-addr <node-id> <link tag-name> <address>
set-addr      0 link1 192.1.1.1
set-addr      0 link2 192.1.2.1
set-addr      2 link1 192.1.5.2
set-addr      2 link2 192.1.6.2
set-addr      3 link1 192.1.7.3
set-addr      3 link2 192.1.8.3

#disable LLT broadcasts
set-bcasthb    0
set-arp        0
```

The `llttab` file on Node 0 would resemble:

```
set-node Node0
set-cluster 1

link link1 udp - udp 50000 - 192.1.1.1 -
link link2 udp - udp 50001 - 192.1.2.1 -

#set address of each link for all peer nodes in the cluster
#format: set-addr <node-id> <link tag-name> <address>
set-addr      1 link1 192.1.3.1
set-addr      1 link2 192.1.4.1
set-addr      2 link1 192.1.5.2
set-addr      2 link2 192.1.6.2
set-addr      3 link1 192.1.7.3
set-addr      3 link2 192.1.8.3

#disable LLT broadcasts
```

```
set-bcasthb      0
set-arp          0
```

Issues and limitations

Several issues and limitations apply:

VCW does not support configuring broadcasting for UDP

The Cluster Configuration Wizard (VCW) does not provide options to configure broadcasting information for UDP. You can configure broadcasting for UDP by manually editing the `llttab` file, as described in this appendix.

If the network adapters are unable to ping each other, the cluster nodes may not get GAB membership

While configuring LLT over UDP, if the network adapters selected for the LLT communication are unable to ping each other and you proceed with the cluster configuration, VCW configures the LLT service on the selected network adapters but the cluster nodes may not receive GAB membership and as a result the Veritas High Availability Engine, HAD, may fail to start.

You can confirm the GAB membership by running the following command:

```
C:\gabconfig -a
```

If no port membership information is returned it indicates that GAB is not operating.

This issue can be addressed in either of the following methods:

- Method 1

Reboot the cluster nodes that do not have GAB membership. Verify the GAB operation in the cluster. Type the following on the command prompt:

```
C:\gabconfig -a
```

If GAB membership information is displayed for all cluster nodes, GAB is working correctly. However, if the appropriate GAB membership information is not returned for one or more nodes, GAB is not operating correctly. In that case, proceed to the next method.

- Method 2

Stop the LLT service in the cluster. Type the following on the command prompt:

```
C:\net stop llc
```


Delete the cluster using VCW.

Ensure that the network adapters are able to ping each other and then re-create the cluster using VCW.

Handling concurrency violation in any-to-any configurations

This appendix includes the following topics:

- [About handling concurrency violation](#)
- [Concurrency violation scenario](#)
- [About the vcsgensvc.vbs script](#)
- [Sample configuration to handle concurrency violation](#)

About handling concurrency violation

This topic describes how you can use the Process agent, along with a sample script, to handle potential concurrency violation in an Any-to Any configuration that uses the GenericService agent.

Concurrency violation scenario

Consider the following excerpt from a configuration in which an antivirus software is configured as a GenericService resource in an Exchange service group.

Exchange Service Group 1

```
System List = {S1, S3}  
GenericService G1 controlling "NortonAntivirus"  
ExchServices E1 managing Exchange Information Store,
```

```
Message Transfer Agent, and System Attendant services
Lanman ExchVName1 controlling the E1 Virtual server
...

G1 requires E1
E1 requires ExchVName1
...
```

Exchange Service Group 2:

```
System List = {S2, S3}
GenericService G2 controlling "NortonAntivirus"
ExchServices E2 managing Exchange Information Store,
Message Transfer Agent, and System Attendant services
Lanman ExchVName2 controlling the E2 Virtual server
...

G2 requires E2
E2 requires ExchVName2
...
```

In this example, S3 is the standby system for both S1 and S2. Service Group 1 is online on S1 and Service Group 2 is online on S2. Such a configuration is desirable for an application like Exchange that requires an antivirus service instance attached to it.

Consider a scenario where Service Group 1 fails over from S1 to S3. When G1 comes online on S3, G2 also reports online because both G1 and G2 monitor the same service. As a result, VCS reports a concurrency violation for Service Group 2 from S3 and tries to take G2 offline on S3. As soon as G2 is taken offline on S3, G1 reports a fault, and Service Group 1 faults on S3.

This situation can be addressed by using a custom script along with the Process agent, in place of the GenericService agent.

About the vcsgensvc.vbs script

The script `vcsgensvc.vbs` resides at the path `%VCS_HOME%\Samples\Process`. The script works with the Process agent to bring services online, monitor them, and take them offline.

[Table E-1](#) depicts the `vcsgensvc.vbs` script parameters.

Table E-1 vcsgensvc.vbs script parameters

Parameter	Accepted Values
Operation	online offline monitor
Service Name	Display or key name of the service
Computer Name	The name of the computer (virtual computer being monitored This parameter applies only to the MonitorProgram attribute of the Process resource.

Sample configuration to handle concurrency violation

In the following sample configuration, the Process agent monitors the Norton Antivirus service. The script is installed in the following directory:

```
D:\Program Files\Veritas\Cluster Server\Samples\Process
```

The script takes the Exchange virtual server name as an input parameter and monitors the service using this virtual name. If the Exchange Virtual server is online and the antivirus service is running, the script returns ONLINE, instead of returning ONLINE based on the status of the service alone.

Note that this recommendation is for this specific scenario only. The Process agent is not an alternative to the GenericService agent, which offers added functionality for generic services.

For Exchange Service Group 1:

```
System List = {S1, S3}
Process AVService1 controlling "NortonAntivirus"
ExchServices E1
Lanman ExchVName1 controlling the E1 Virtual server
...

G1 requires E1
E1 requires ExchVName1
...

Lanman L1 (
    VirtualName = ExchVName1
```

```

)

Process AVService1 (
  StartProgram = "CScript.exe \"d:\\program files\\
  veritas\\cluster
  server\\samples\\process\\vcsgensvc.vbs\"online
  NortonAntivirus"
    StopProgram = "CScript.exe \"d:\\program
  files\\veritas\\cluster
  server\\samples\\process\\vcsgensvc.vbs\" offline
  NortonAntivirus"
  MonitorProgram = "CScript.exe \"d:\\program files\\
  veritas\\cluster server\\samples\\process\\vcsgensvc.vbs\"
  monitor NortonAntivirus ExchVName1"
)

```

For Exchange Service Group 2:

```

System List = {S2, S3}
Process AVService2 controlling "NortonAntivirus"
ExchServices E2
Lanman ExchVName2 controlling the E2 Virtual server
...

G2 requires E2
E2 requires ExchVName2
...

Lanman L2 (
  VirtualName = ExchVName2
)

Process AVService2 (
  StartProgram = "CScript.exe \"d:\\program files\\
  veritas\\cluster server\\samples\\process\\vcsgensvc.vbs\"
  online NortonAntivirus"
    StopProgram = "CScript.exe \"d:\\program files\\
  veritas\\cluster server\\samples\\process\\vcsgensvc.vbs\"
  offline NortonAntivirus"
  MonitorProgram = "CScript.exe \"d:\\program files\\
  veritas\\cluster server\\samples\\process\\vcsgensvc.vbs\"
  monitor NortonAntivirus ExchVName2"
)

```

Notes for using scripts with the Process agent

Following are some requirements that you must follow when using scripts with the Process agent:

- In the above example, the supplied script assumes that Service Group 1 and Service Group 2 will never come online on one system. Service Group Workload Management or triggers must be configured to meet this requirement.
- When using this configuration, we recommend setting the user context of the Process agent to LocalSystem.

Accessibility and VCS

This appendix includes the following topics:

- [About accessibility in VCS](#)
- [Navigation and keyboard shortcuts](#)
- [Support for accessibility settings](#)
- [Support for assistive technologies](#)

About accessibility in VCS

Cluster Server provides shortcuts for major graphical user interface (GUI) operations and menu items. Cluster Server is compatible with operating system accessibility settings as well as a variety of assistive technologies. All manuals also are provided as accessible PDF files, and the online help is provided as HTML, which appears in a compliant viewer.

Navigation and keyboard shortcuts

VCS uses standard operating system navigation keys and keyboard shortcuts. For its unique functions, VCS uses its own navigation keys and keyboard shortcuts which are documented below.

Navigation in the Java Console

[Table F-1](#) lists keyboard navigation rules and shortcuts used in Cluster Manager (Java Console), in addition to those provided by the operating system.

Table F-1 Keyboard inputs and shortcuts

VCS keyboard input	Result
[Shift F10]	Opens a context-sensitive pop-up menu
[Spacebar]	Selects an item
[Ctrl Tab]	Navigates outside a table
[F2]	Enables editing a cell

Navigation in the Web console

The Web console supports standard browser-based navigation and shortcut keys for supported browsers.

All Veritas GUIs use the following keyboard navigation standards:

- Tab moves the cursor to the next active area, field, or control, following a preset sequence.
- Shift+Tab moves the cursor in the reverse direction through the sequence.
- Up-arrow and Down-arrow keys move the cursor up and down the items of a list.
- Either Enter or the Spacebar activates your selection. For example, after pressing Tab to select Next in a wizard panel, press the Spacebar to display the next screen.

Support for accessibility settings

Veritas software responds to operating system accessibility settings.

Veritas products are compatible with accessibility utilities provided by operating systems.

On UNIX systems, you can change the accessibility settings by using desktop preferences or desktop controls.

On Windows systems, you can set accessibility options that involve keyboard responsiveness, display contrast, alert sounds, and mouse operation through the Control Panel (**Start > Settings > Control Panel > Accessibility Options**) and through the Accessibility Wizard (**Start > All Programs > Accessories > Accessibility > Accessibility Wizard**). On Windows 2012 operating systems, open the **Start** screen and type in the title of the UI that you want to launch; Windows searches the string you entered and lists the available UIs.

Support for assistive technologies

Veritas provides support for assistive technologies as follows:

- Cluster Manager (Java Console) is compatible with JAWS 4.5.
- Though graphics in the documentation can be read by screen readers, setting your screen reader to ignore graphics may improve performance.
- Veritas has not tested screen readers for languages other than English.