VERITAS Cluster Server Agents for VERITAS Volume Replicator 4.1

Configuration Guide

HP-UX

N13013G

June 2005

VERITAS Software Corporation
350 Ellis Street
Mountain View, CA 94043
USA
Phone 650–527–8000 Fax 650–527–2908
www.veritas.com

Contents

Preface

Agents monitor specific resources within an application, determine the status of these resources, and start or stop them according to external events. These processes are common to all agents, but how they are performed depends on the resource being monitored. Agents are add-on applications.

Note If this document is dated more than six months prior to the date you are installing the agents, contact VERITAS™ Customer Support to confirm the latest supported versions of the application and operating system.

How This Guide Is Organized

Chapter 1. "Overview of the VCS Agents for VVR" on page 1 explains how the VCS agents for VVR work and gives an overview of how to set up VERITAS Volume Replicator (VVR) in a VCS environment.

Chapter 2. "Configuring the Agents for High Availability" on page 17 describes how to create a VVR configuration and how to place VVR under VCS control for high availability.

Related Documentation

For more information on any of the topics presented in this guide, refer to the VERITAS Cluster Server, VERITAS Volume Manager, and VERITAS Volume Replicator documentation sets.

Conventions

Convention	Usage	Example
`monospace`	Used for path names, commands, output, directory and file names, functions, and parameters.	Read tunables from the `/etc/vx/tunefstab` file. See the `ls`(1) manual page for more information.
`monospace` (**`bold`**)	Indicates user input.	# **`ls pubs`** C:\> **`dir pubs`**
italic	Identifies book titles, new terms, emphasized text, and variables replaced with a name or value.	See the *User's Guide* for details. The variable *system_name* indicates the system on which to enter the command.
bold	Depicts GUI objects, such as fields, list boxes, menu selections, etc. Also depicts GUI commands.	Enter your password in the **Password** field. Press **Return.**
blue text	Indicates hypertext links.	See "Getting Help" on page ix.
#	Unix superuser prompt (all shells).	# **`cp /pubs/4.0/user_book /release_mgnt/4.0/archive`**

C:\>	Windows user prompt.	C:\> **`copy \pubs\4.0\user_book`**
		`c:\release_mgnt\4.0\archive`

Getting Help

For technical assistance, visit http://support.veritas.com and select phone or email support. This site also provides access to resources such as TechNotes, product alerts, software downloads, hardware compatibility lists, and our customer email notification service. Use the Knowledge Base Search feature to access additional product information, including current and past releases of VERITAS documentation.

For license information, software updates and sales contacts, visit https://my.veritas.com/productcenter/ContactVeritas.jsp. For information on purchasing product documentation, visit http://webstore.veritas.com.

Documentation Feedback

Your feedback on product documentation is important to us. Send suggestions for improvements and reports on errors or omissions to vvrdocs@veritas.com. Include the title and part number of the document (located in the lower left corner of the title page), and chapter and section titles of the text on which you are reporting. Our goal is to ensure customer satisfaction by providing effective, quality documentation. For assistance with topics other than documentation, visit http://support.veritas.com.

Overview of the VCS Agents for VVR 1

Agents are processes that manage predefined resource types. When an agent is started, it obtains configuration information from the VERITAS Cluster Server (VCS). It then periodically monitors the resources and updates VCS with the resource status. Typically agents do the following:

- Bring resources online
- Take resources offline
- Monitor resources and report any state changes to VCS

List of the VCS Agents for VVR

The VCS Agents for VVR monitor and manage Replicated Volume Groups (RVGs). Each agent includes VCS-type declarations and agent executables, which represent a resource type. The VCS Agents for VVR include:

- RVG Agent
- RVGPrimary Agent
- RVGSnapshot Agent

A description of each agent begins on "How the Agents Work" on page 3.

VCS Cluster Components

Resources, attributes, and service groups are components integral to cluster functionality. For more information, see the *VERITAS Cluster Server User's Guide*.

Resources

Resources are hardware or software entities, such as disks, volumes, file system mount points, network interface cards (NICs), IP addresses, applications, and databases. Resources work together to provide a service to clients in a client/server environment. The bundled agents resource types are defined in the `types.cf` file by a collection of attributes. The VCS configuration file, `main.cf`, contains the values for the attributes of the resources. The `main.cf` file incorporates the resources listed in the `types.cf` by way of an `include` directive. The `main.cf` file also incorporates the VVR resource types, which are defined in the file `VVRTypes.cf` by way of an `include` directive.

Attributes

Attributes contain data regarding the cluster, nodes, service groups, resources, resource types, and agents. A specified value for a given attribute configures the resource to function in a specific way. By modifying the value of an attribute of a resource, you change the way the VCS agent manages the resource. Each attribute has a definition and a value. You define an attribute by specifying its data type and dimension. Attributes also have default values that are assigned when a value is not specified.

Service Groups

Service groups are comprised of related resources. When a service group is brought online, all the resources within the group are brought online.

Modifying the Agents and Their Resources

You can use the VCS commands from the command line to modify the configuration of the resources managed by an agent. You can also edit the `main.cf` file directly, however you must stop VCS before editing the `main.cf` file. Example `main.cf` files for the VCS Agents for VVR are located in the `/etc/VRTSvcs/conf/sample_vvr/RVG` directory.

How the Agents Work

This section describes how each agent works, summarizes the entry points, state definitions, and attributes for each agent, and explains the dependency graphs for each agent.

The VCS Agents for VVR include:

- RVG Agent
- RVGPrimary Agent
- RVGSnapshot Agent

RVG Agent

The RVG agent enables replication between clusters by managing the Primary VVR node in one cluster and the Secondary VVR node in another cluster, each of which can be failed over in its respective cluster. In this way, replication is made highly available.

Note The RVG works with the RVGPrimary agent to provide failover of the Primary VVR node to the Secondary VVR node. If a disaster occurs on the Primary VVR node and all the nodes in the Primary cluster are unavailable, the RVG agent does not fail over the Primary role from the Primary VVR node to the Secondary VVR node. The Global Cluster Option of VCS enables you to fail over the Primary role from a Primary VVR node to a Secondary VVR node.

The RVG agent includes the following key features:

- Removes potential single points of failure by enabling Primary and Secondary VVR nodes to be clustered.
- Makes the process of starting VCS-managed applications that use VVR, as easy as bringing a VCS service group online.
- Continues replication after a node in a cluster fails without losing updates.
- Ensures that VVR can be added to any VCS cluster by including the RVG resource type definitions.

An example configuration file for this agent that can be used as a guide when creating your configuration is located at `/etc/VRTSvcs/conf/sample_vvr/RVG`.

Note The attributes `Primary`, `SRL`, `RLinks` of the RVG agent are optional in this release, that is, the RVG agent no longer requires the optional attributes to configure the RVG resources. The RVG agent will *not* contain these optional attributes in the next release; configurations that use the optional attributes will fail in the next release.

The following table summarizes the function of the RVG agent, its entry points, state definitions, and attributes:

Description	Brings the RVG online, monitors read/write access to the RVG, and takes the RVG offline; this is a failover resource.	
Entry Points	online—Verifies whether the DiskGroup agent has recovered the RVG. If not, recovers and starts the data volumes and the Storage Replicator Log (SRL), recovers the RVG, recovers all RLINKs in the RVG, and then starts the RVG. offline—Stops the RVG. clean—Stops the RVG. info—Gives the information about the replication status for the Replicated Data Set (RDS). monitor—Monitors the state of the RVG using the `vxprint` command. **Note** The RVG resource monitors an RVG for local access only; it does not monitor replication.	
Detecting Failure	The RVG resource fails if the RVG is not in the `ENABLED/ACTIVE` state.	
State Definitions	ONLINE—Indicates that the RVG is in ENABLED/ACTIVE state. OFFLINE—Indicates that the RVG is in DISABLED/CLEAN state.	
Required Attributes	**Type and Dimension**	**Definition**
RVG	string-scalar	The name of the RVG being monitored.
DiskGroup	string-scalar	The disk group with which this RVG is associated.
Optional Attributes	**Type and Dimension**	**Definition**
Primary	string-scalar	Indicates Primary RVG (true, false).
SRL	string-scalar	The SRL associated with this RVG.
RLinks	string-vector	The list of RLINKs associated with the RVG.

Type Definition

```
type RVG (
  static str ArgList[] = { RVG, DiskGroup, Primary, SRL, RLinks}
  str RVG
  str DiskGroup
  str Primary
  str SRL
  str RLinks[]
  static int NumThreads = 1
)
```

Using the info Entry Point

The info entry point displays information about the replication status of an RDS. By default, the info interval is set to zero. To change the default info interval, use the following command:

```
# hatype -modify resourcetype_name InfoInterval interval
```

For example, to set the info interval to 60 seconds for the RVG resource type, enter:

```
# hatype -modify RVG InfoInterval 60
```

The info interval indicates how frequently VCS executes the info entry point to update the the replication status. In the above example, the info interval is set to 60, so VCS updates the replication status every 60 seconds. To display the output of the info entry point, use the following command:

```
# hares -value resource_name ResourceInfo
```

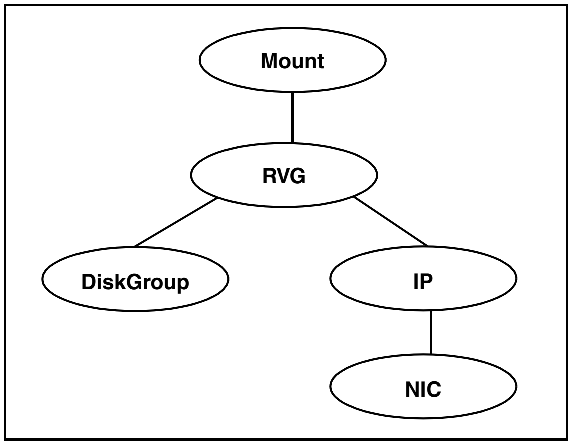
The output of the info entry point is also logged in the file `/var/VRTSvcs/log/engine_A.log`.

Dependency Graph for the RVG Agent

The RVG resource represents the RVG (Replicated Volume Group) in the RDS (Replicated Data Set). The RVG resource is dependent on the DiskGroup resource. The RVG resource is also dependent on the IP resources that it uses for replication.

In a VVR environment, higher-level application resources, such as Mount, that would typically depend on a Volume resource must depend on the associated RVG resource. Refer to the *VERITAS Cluster Server User's Guide* for more information on dependencies.

Dependency Graph for the RVG Agent



RVGPrimary Agent

The RVGPrimary agent enables migration and takeover of a VVR replicated data set in a VCS environment. Bringing a resource of type RVGPrimary online causes the RVG on the local host to become a primary if it not already. The agent is useful when hosts in both the primary and secondary side are clustered, in particular a VCS replicated data cluster or when using the Global Cluster Option, to completely automate the availability of writable replicated disks to an application managed by VCS.

The RVGPrimary agent includes the following key features:

- Removes manual steps of migrating a VVR primary and secondary roles when failing over applications across a wide area.
- Minimizes the need for resynchronizing replicated volumes by attempting a migration before attempting a hard takeover.
- Waits for the two sides of a replicated data set to become completely synchronized before migrating roles.
- Supports an automatic fast failback resynchronization of a downed primary if it later returns after a takeover.

A sample configuration file for this agent that can be used as a guide when creating your configuration is located at `/etc/VRTSvcs/conf/sample_vvr/RVGPrimary`. For specifics about this configuration see the *VERITAS Cluster Server User's Guide.*

The following table summarizes the function of the RVGPrimary agent, its entry points, state definitions, and attributes:

Description	Attempts to migrate or takeover a Secondary to a Primary upon an application failover.	
Entry Points	Online—Determines the current role of the RVG; if Secondary, attempt a migrate, waiting for any outstanding writes from the original Primary; if the original Primary is down attempt a takeover; if the RVG is a Primary, perform no actions and go online Offline—Perform no actions. Clean—Perform no actions. Monitor—Perform no actions; monitoring of the actual RVG is done by the RVG agent.	
	Detecting Failure Monitoring of the actual RVG is done by the RVG agent; accidental migration of a VVR Primary outside of VCS would cause other resources to fault immediately, such as Mount, so no special monitoring by this agent is necessary.	
Attributes	**Type and Dimension**	**Definition**
RvgResourceName	string-scalar	The name of the RVG resource type that this agent will promote, that is, the name RVG resource type which has been configured using the RVG agent.
AutoTakeover	integer-scalar	A flag to indicate whether the agent should perform a takeover on online if the original Primary is down.
AutoResync	integer-scalar	A flag to indicate whether the agent should attempt to automatically perform a fast-failback resynchronization of the original Primary after a takeover and after the original Primary returns.

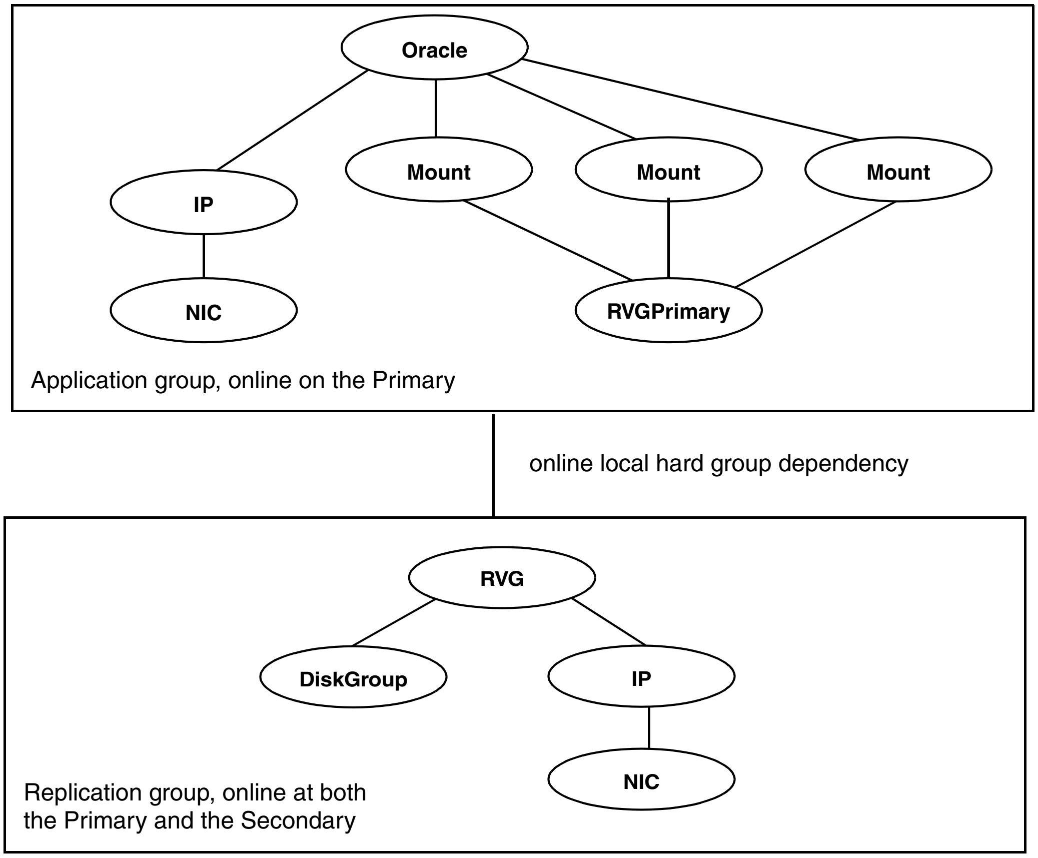
Type Definition

```
type RVGPrimary (
 static keylist SupportedActions = { fbsync }
 static int InfoTimeout = 0
 static int NumThreads = 1
 static int OnlineRetryLimit = 1
 static str ArgList[] = { RvgResourceName, AutoTakeover, AutoResync }
 str RvgResourceName
 int AutoTakeover = 1
 int AutoResync = 0
)
```

Dependency Graph for the RVGPrimary Agent

The RVGPrimary agent is customarily used in conjunction with the RVG agent in two groups with an online local firm group dependency; the parent group contains the resources managing the actual application and file systems as well as the RVGPrimary resource, and the child group contains the resources managing the storage infrastructure, including the RVG and DiskGroup type resources. Refer to the *VERITAS Cluster Server User's Guide* for more information on detailed setup of a VVR environment using the RVGPrimary agent.

Dependency Graph for the RVGPrimary Agent


Oracle
IP
Mount
Mount
Mount
NIC
RVGPrimary
Application group, online on the Primary
online local hard group dependency
RVG
DiskGroup
IP
NIC
Replication group, online at both the Primary and the Secondary

RVGSnapshot Agent

The RVGSnapshot agent automates the taking of space-optimized snapshots on a secondary RVG; since these snapshots can be mounted and written to without affecting the actual replicated data, a space-optimized snapshot can be an effective tool for scheduling a "fire drill" to confirm that a wide-area failover is possible. By combining this agent with VCS Mount agents and VCS agents that manage the application being replicated, a special fire drill service group can be created that can be onlined and offlined at regularly scheduled intervals to confirm the robustness of a disaster recovery environment.

In addition to the agent itself, a text-based wizard `/opt/VRTSvcs/bin/fdsetup` that prepares the VVR and VCS infrastructure for a fire drill and a script `/opt/VRTSvcs/bin/fdsched` that runs the fire drill and consolidates the results are included with this package. Complete details are in the *VERITAS Cluster Server User's Guide*.

The RVGSnapshot agent includes the following key features:

- Automates the process of creating a space-optimized snapshot on a VVR secondary that can be mounted to simulate a wide-area failover without affecting the production application.
- Includes a wizard to effectively set up and schedule fire drills that are completely managed by VCS.

While the `fdsetup` wizard configures the appropriate resources for a fire drill group, the following table summarizes the function of the RVGSnapshot agent, its entry points, state definitions, and attributes:

Description	Creates and destroys a transactionally consistent space-optimized snapshot of all volumes in a VVR secondary replicated data set.	
Entry Points	online—Creates a transactionally consistent snapshot of all volumes in the RDS. offline—Destroys the snapshot. clean—Cleans up any failed snapshot creation or deletion. monitor—No operation; failure of the snapshot will be indicated by the failure of the Mount resource of any filesystems mounted on it.	
Detecting Failure	The RVGSnapshot resource faults on timeout if a snapshot creation did not succeed during an online.	
State Definitions	ONLINE—Indicates that a snapshot was created. OFFLINE—Indicates that a snapshot was destroyed.	
Required Attributes	**Type and Dimension**	**Definition**
RvgResource Name	string-scalar	The name of the VCS RVG-type resource that manages the RVG that will be snapshot by this agent.
CacheObj	string-scalar	Name of the cache object that is required for a space-optimized snapshot; the `fdsetup` wizard will create one if it does not exist
Prefix	string-scalar	Token prepended to the name of the actual volume when creating the snapshotted volumes.
Optional Attributes	**Type and Dimension**	**Definition**
DestroyOnOf fline	int-scalar	A flag to indicate whether to destroy the snapshot upon offlining the resources. For a fire drill, the snapshot should be deleted to reduce any performance impact of leaving the snapshot for a long period of time; however, if there is interest in keeping the data, then this value should be set to 0. The default is 1 (true).
FDFile	temporary string-scalar	The fire drill schedule updates this attribute with the system name and the path to a file containing the output of the last complete fire drill for the group containing an RVGSnapshot resource.

Type Definition

```
type RVGSnapshot (
 static keylist RegList = { Prefix }
 static int InfoTimeout = 0
 static int NumThreads = 1
 static str ArgList[] = { RvgResourceName, CacheObj, Prefix,
DestroyOnOffline }
 str RvgResourceName
 str CacheObj
 str Prefix
 boolean DestroyOnOffline = 1
 temp str FDFile
)
```

How the Agents for Hybrid Applications Work

The agents for hybrid applications include:

- RVG Agent
- RVGPrimary Agent

A hybrid configuration is for Replicated Data Clusters (RDCs) and is a combination of the failover and parallel service groups. A hybrid service group behaves like a failover group within a system zone and like a parallel group across system zones. It cannot fail over across system zones. A switch operation on a hybrid service group is allowed only between systems within the same system zone.

For more information about the RVG agent and RVGPrimary agent, see "RVG Agent" on page 3 and "RVGPrimary Agent" on page 6 respectively. These section give information about the entry points, state definitions, and attributes for the RVG agent and the RVGPrimary agent. In addition, the following attribute must be set for the RVG agent and the RVGPrimary agent while configuring RDCs:

Optional Attributes	**Type and Dimension**	**Definition**
SystemZones	integer-association	Indicates failover zone.

An RDC uses VVR as opposed to shared storage to provide access to data at the Secondary. An RDC exists within a single VCS cluster. The application group, which is configured as a failover group, can be online only on the Primary host. In the case of the failure of the Primary site, the Secondary is promoted to a Primary and the application is brought online on the new Primary host.

An RDC configuration is appropriate in configurations lacking shared storage or SAN interconnection between the Primary site and Secondary site, but where dual dedicated LLT links are available between the Primary site and the Secondary site. For more information about RDCs, refer to the *VERITAS Cluster Server User's Guide*.

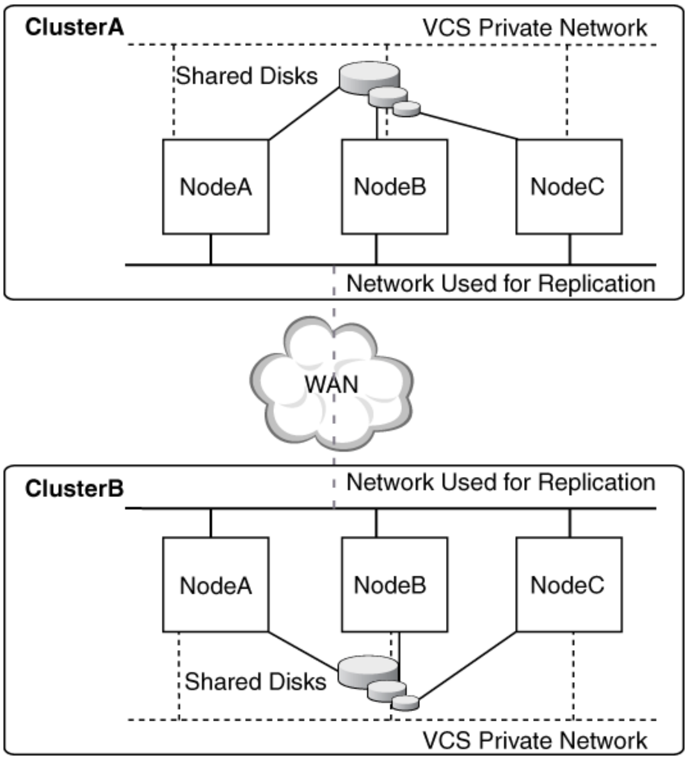
Overview of how to Configure VVR in a VCS Environment

This section gives an overview of how to configure VVR in a VCS environment for high availability of the application that is involved in replication. To configure VVR in a VCS environment, you must perform the following tasks:

1. Setting up a VVR configuration, which involves creating a Replicated Data Set (RDS).
2. Creating service groups for the VVR agents and adding the resource and group dependencies appropriately.

Generic VVR Setup in a VCS Environment

The following illustration shows how VVR replicates in a VCS environment given a two-cluster environment.

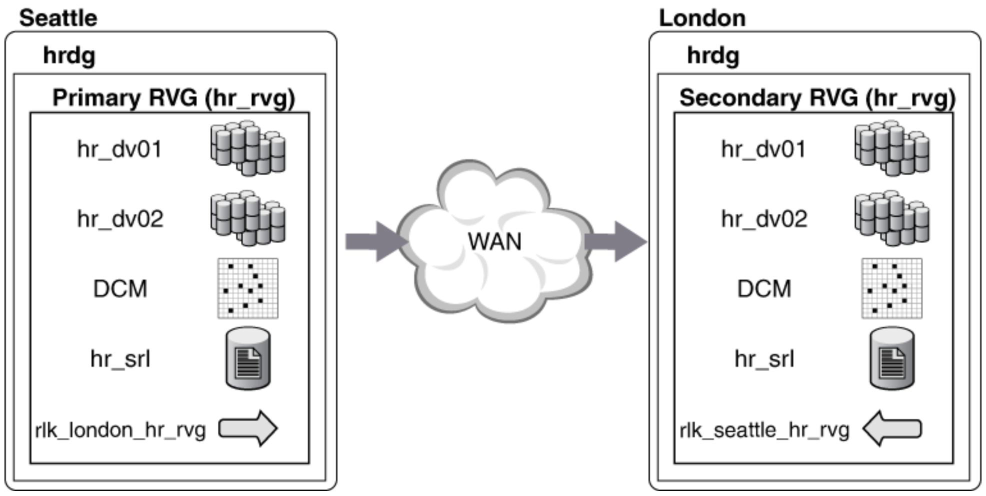


Example VVR Configuration in a VCS Environment

In the following example, two clusters are located at separate sites. VVR replicates data between the sites using a WAN.

The first cluster is located in Seattle and is named `Seattle`. The cluster `Seattle` consists of two nodes: `seattle1` and `seattle2`. The second cluster is located in London and is named `London`.The cluster `London` also consists of two nodes: `london1` and `london2`. The nodes located in the cluster `Seattle` contain the Primary RVG. The nodes located in the cluster `London` contain the Secondary RVG. Note that the following illustration shows the names of the VVR components used by the RVG agent.

Example—VVR Configuration in a VCS Environment



Configuring the Agents for High Availability 2

This chapter describes the requirements and best practices for creating service groups containing (Replicated Volume Groups) RVGs involved in replication. It also explains how to create a VVR configuration and how to place VVR under VCS control for high availability.

Requirements for Configuring VVR in a VCS Environment

- Each node that is part of a particular VCS service group involved in replication must use the same port number for replication. You may need to change this number on some nodes before configuring VVR.
- If a node has more than one network interface card on the same physical network being used for replication, each network interface card must have a different MAC address. This is true for all the nodes at the Primary and Secondary sites.
- This requirement is specific to RVGAgent. VCS requires the `noautoimport` attribute of the disk group to be set. Perform the following steps before starting the VCS cluster:

  a. Check whether the `noautoimport` attribute is set by issuing the following command:

  ```
  # vxprint -l diskgroup
  ```

  If the output displays the `noautoimport` attribute in the `info` field, omit the following step and proceed with cluster services.

  b. To set the `noautoimport` attribute, enter the following commands:

  ```
  # vxdg deport diskgroup
  # vxdg -t import diskgroup
  ```

  To verify that the `noautoimport` attribute is set, issue the command as described in step a.

Best Practices for Setting Up the Agents

- Only one `DiskGroup` and one `RVG` resource must be present in a service group.
- If a disk group is configured as a `DiskGroup` resource, then all the RVGs in this disk group must be configured as `RVG` resources.
- When configuring single-instance failover applications, use the RVG, RVGPrimary, and RVGSnapshot agents.

Best Practices for Setting Up Replication

Refer to the *VERITAS Volume Replicator Administrator's Guide*.

Adding the VVR Agents to the VCS Configuration

This section explains how to add the VVR agents to the VCS configuration:

- When VCS is running
- When VCS is stopped

When VCS is Running

To add the agents without stopping the applications on a system, log in as `root` on one node in the cluster and perform the following steps:

1. Set the configuration mode to read/write by typing the following command on any system in the cluster:

   ```
   # haconf -makerw
   ```

2. Automatically copy the `VVRTypes.cf` file from the `/etc/VRTSvcs/conf` directory to `/etc/VRTSvcs/conf/config` directory and include the `VVRTypes.cf` file to the existing configuration file `main.cf` by running the following script:

   ```
   # /etc/VRTSvcs/conf/sample_vvr/RVG/addVVRTypes.sh
   ```

3. Ensure that all changes to the existing configuration have been saved and that further changes are prevented.

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

4. If you stopped the agent before installing the new agent, start the agent on the system by entering:

   ```
   # haagent -start agent_name -sys system_name
   ```

   When you get the message `Please look for messages in the log file`, check the file `/var/VRTSvcs/log/engine_A.log` for a message confirming that each agent has started.

   You can also use the `ps` command to confirm that the agent is started.

5. If you brought the RVG service group offline before doing the installation, bring it online by using the following command:

   ```
   # hagrp -online service_group -sys system_name
   ```

When VCS is Stopped

You can add the agents by editing the `main.cf` file. You must stop VCS before editing the `main.cf` file. Log in as `root` on one node in the cluster and perform the following steps:

1. Ensure that all changes to the existing configuration have been saved and that further changes are prevented while you modify `main.cf` located in the `/etc/VRTSvcs/conf/config` directory.

   If the VCS cluster is currently writeable, run the following command:

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

   If the VCS cluster is already read only, run the following command:

   ```
   # haconf -dump
   ```

2. Do not edit the configuration files while VCS is running. The following command stops the `had` daemon on all systems and leaves resources available:

   ```
   # hastop -all -force
   ```

3. Copy the `VVRTypes.cf` file from `/etc/VRTSvcs/conf` to the `/etc/VRTSvcs/conf/config` directory.

4. Add the VVRTypes to the `main.cf` file, located in `/etc/VRTSvcs/conf/config` directory.

   For a new agent installation, add the following line to the `main.cf` file:

   ```
   include “VVRTypes.cf”
   ```

5. Verify the syntax of the file `/etc/VRTSvcs/conf/config/main.cf`:

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

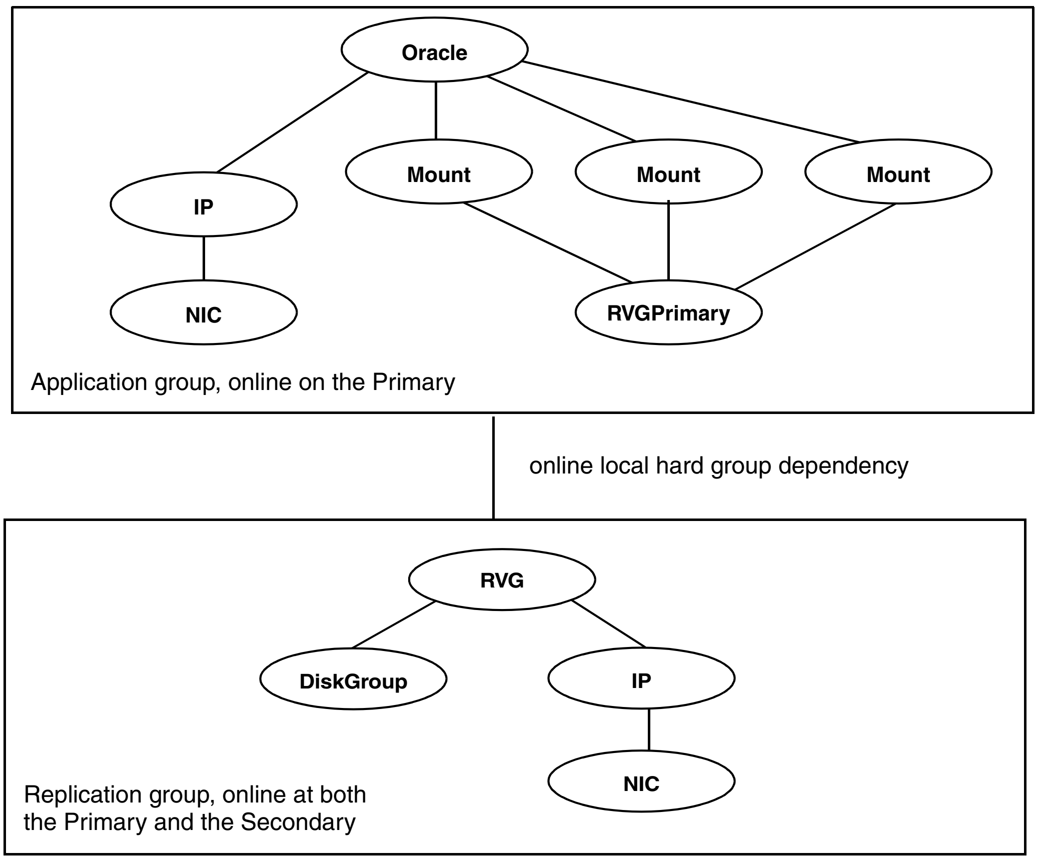
6. Start the VCS engine:

   ```
   # hastart
   ```

Example Configuration for a Failover Application

In the following example, a failover application that uses an RVG is made highly available across two clusters. The application service group contains the following resources: application, Mount, NIC, IP, and RVGPrimary. The replication group contains the RVG, IP, NIC, and DiskGroup resources. The application group has an online local hard dependency on the replication group.

RVG and RVGPrimary Agents—Service Groups and Resource Dependencies


Oracle
Mount
Mount
Mount
IP
NIC
RVGPrimary
Application group, online on the Primary
online local hard group dependency
RVG
DiskGroup
IP
NIC
Replication group, online at both the Primary and the Secondary

Example—Setting Up VVR in a VCS Environment

Configuring VVR with VCS requires the completion of several tasks, each of which must be performed in the order presented below.

- ✔ Setting Up the VVR Configuration
- ✔ Verifying the VVR Replication State
- ✔ Configuring the Agents

Before setting up the VVR configuration, verify whether all the nodes in the cluster that have VVR installed use the same port number for replication. To verify and change the port numbers, use the `vrport` command. For instructions on using the `vrport` command, see the *VERITAS Volume Replicator Administrator's Guide*. If the port number is the same on all nodes, add the VVR agents to the VCS configuration.

Setting Up the VVR Configuration

The example in this section refers to the sample configuration shown in "Example VVR Configuration in a VCS Environment" on page 16. Note that the VVR configuration that is being set up in this example applies to the RVG Agent, that is, it uses the names that are used in the sample configuration file of the RVG agent.

The procedure to configure VVR is the same for all the VVR agents. Use the sample configuration files located in `/etc/VRTSvcs/conf/sample_vvr/RVG` directory to configure the other agents. For more information on configuring VVR, refer to the *VERITAS Volume Replicator Administrator's Guide*. The example uses the names listed in the following table.

Name of Cluster: `Seattle`

Disk group	`hrdg`
Primary RVG	`hr_rvg`
Primary RLINK to london1	`rlk_london_hr_rvg`
Primary data volume #1	`hr_dv01`
Primary data volume #2	`hr_dv02`
Primary SRL for hr_rvg	`hr_srl`
Cluster IP	`10.216.144.160`

Name of Cluster: London

Disk group	hrdg
Secondary RVG	hr_rvg
Secondary RLINK to seattle	rlk_seattle_hr_rvg
Secondary data volume #1	hr_dv01
Secondary data volume #2	hr_dv02
Secondary SRL for hr_rvg	hr_srl
Cluster IP	10.216.144.162

This example assumes that each of the hosts seattle1 and london1 has a disk group named hrdg with enough free space to create the VVR objects mentioned in the example. Set up the VVR configuration on seattle1 and london1 to include the objects used in the sample configuration files, main.cf.seattle and main.cf.london, located in the /etc/VRTSvcs/conf/sample_vvr/RVG directory.

1. On london1:

   a. Create the Secondary data volumes.

   ```
   # vxassist -g hrdg make hr_dv01 100M \
      layout=mirror logtype=dcm mirror=2

   # vxassist -g hrdg make hr_dv02 100M \
      layout=mirror logtype=dcm mirror=2
   ```

   b. Create the Secondary SRL.

   ```
   # vxassist -g hrdg make hr_srl 200M mirror=2
   ```

2. On seattle1:

   a. Create the Primary data volumes.

   ```
   # vxassist -g hrdg make hr_dv01 100M \
      layout=mirror logtype=dcm mirror=2

   # vxassist -g hrdg make hr_dv02 100M \
      layout=mirror logtype=dcm mirror=2
   ```

   b. Create the Primary SRL.

   ```
   # vxassist -g hrdg make hr_srl 200M mirror=2
   ```

c. Create the Primary RVG.

```
# vradmin -g hrdg createpri hr_rvg \
     hr_dv01,hr_dv02 hr_srl
```

d. Determine the virtual IP address to be used for replication, and then verify that the device interface for this IP is plumbed. If the device interface for this IP is not plumbed, then plumb the device. Get the IP up using the OS-specific command. This IP address that is to be used for replication must be configured as the IP resource for this RVG service group.

e. Create the Secondary RVG.

```
# vradmin -g hrdg addsec hr_rvg \
   10.216.144.160 10.216.144.162 prlink=rlk_london_hr_rvg \
      srlink=rlk_seattle_hr_rvg
```

Note The RLINKs must point to the virtual IP address for failovers to succeed. The virtual IP address 10.216.144.160 must be able to ping virtual IP address 10.216.144.162 and vice versa.

f. Start Replication.

```
# vradmin -g hrdg -f startrep hr_rvg
```

3. Create the following directories on `seattle1` and `seattle2`. These directories will be used as mount points for volumes `hr_dv01` and `hr_dv02` on the `seattle` site.

```
# mkdir /hr_mount01
# mkdir /hr_mount02
```

4. On `seattle1` and `seattle2`, create file systems on the volumes `hr_dv01` and `hr_dv02`.

Verifying the VVR Replication State

Test the replication state between `seattle1` and `london1` to verify that VVR is configured correctly. Type the following command on each node:

```
# vxprint -g hrdg hr_rvg
```

✔ Verify that the state of the RVG is ENABLED/ACTIVE.

✔ Verify that the state of the RLINK is CONNECT/ACTIVE.

Configuring the Agents

This section explains how to configure the VVR agents.

Configuration Tasks

This section gives instructions on how to configure the RVG agent and RVGPrimary agent when VCS is stopped and when VCS is running. Sample configuration files, `main.cf.seattle` and `main.cf.london`, are located in the `/etc/VRTSvcs/conf/sample_vvr/RVG` and `/etc/VRTSvcs/conf/sample_vvr/RVGPrimary` directories respectively, and can be used for reference.

You can add the RVG resource to your existing VCS configuration using any one of the following procedures:

- ✔ Configuring the Agents When VCS is Running
- ✔ Configuring the Agents When VCS is Stopped

Configuring the Agents When VCS is Running

The example in this section explains how to configure the RVG and RVGPrimary agents when VCS is running. For details about the example configuration, see "Example Configuration for a Failover Application" on page 21

Note Use this example as a reference when creating or changing your resources and attributes.

Perform the following steps on the system `seattle1` in the Primary cluster `Seattle`:

1. Log in as root.

2. Set the VCS configuration mode to read/write by issuing the following command:

```
# haconf -makerw
```

3. Create the replication service group, VVRGrp. This group contains all the storage and replication resources.

   a. Add a service group, VVRGrp, to the cluster Seattle and modify the attributes SystemList and AutoStartList of the service group to populate SystemList and AutoStartList:

   ```
   # hagrp -add VVRGrp
   # hagrp -modify VVRGrp SystemList seattle1 0 seattle2 1
   # hagrp -modify VVRGrp AutoStartList seattle1 seattle2
   ```

   b. Add the DiskGroup resource Hr_Dg to the service group VVRGrp and modify the attributes of the resource:

   ```
   # hares -add Hr_Dg DiskGroup VVRGrp
   # hares -modify Hr_Dg DiskGroup hrdg
   ```

   c. Add the RVG resource Hr_Rvg to the service group VVRGrp and modify the attributes of the resource:

   ```
   # hares -add Hr_Rvg RVG VVRGrp
   # hares -modify Hr_Rvg RVG hr_rvg
   # hares -modify Hr_Rvg DiskGroup hrdg
   ```

   d. Add a NIC resource vvrnic to the service group VVRGrp and modify the attributes of the resource:

   ```
   # hares -add vvrnic NIC VVRGrp
   # hares -modify vvrnic Device eth3
   ```

   e. Add the IP resource vvrip to the service group VVRGrp and modify the attributes of the resource:

   ```
   # hares -add vvrip IP VVRGrp
   # hares -modify vvrip Device eth3
   # hares -modify vvrip Address 192.2.40.20
   # hares -modify vvrip NetMask "255.255.248.0"
   ```

   f. Specify resource dependencies for the resources you added in the previous steps:

   ```
   # hares -link Hr_Rvg vvrip
   # hares -link Hr_Rvg Hr_Dg
   # hares -link vvrip vvrnic
   ```

   g. Enable all resources in VVRGrp

   ```
   # hagrp -enableresources VVRGrp
   ```

4. Create the application service group, `ORAGrp`. This group contains all the application specific resources.

   a. Add a service group, `ORAGrp`, to the cluster `Seattle` and populate the attributes `SystemList`, `AutoStartList` and `ClusterList` of the service group

   ```
   # hagrp -add ORAGrp
   # hagrp -modify ORAGrp SystemList seattle1 0 seattle2 1
   # hagrp -modify ORAGrp AutoStartList seattle1 seattle2
   # hagrp -modify ORAGrp ClusterList  Seattle 0 London 1
   ```

   b. Add a NIC resource `oranic` to the service group `ORAGrp` and modify the attributes of the resource:

   ```
   # hares -add oranic NIC ORAGrp
   # hares -modify oranic Device eth0
   ```

   c. Add an IP resource `oraip` to the service group `ORAGrp` and modify the attributes of the resource:

   ```
   # hares -add oraip IP ORAGrp
   # hares -modify oraip Device eth0
   # hares -modify oraip Address 192.2.40.1
   # hares -modify oraip NetMask "255.255.248.0"
   ```

   d. Add the Mount resource `Hr_Mount01` to mount the volume `hr_dv01` in the RVG resource `Hr_Rvg`:

   ```
   # hares -add Hr_Mount01 Mount ORAGrp
   # hares -modify Hr_Mount01 MountPoint /hr_mount01
   # hares -modify Hr_Mount01 BlockDevice \
      /dev/vx/dsk/Hr_Dg/hr_dv01
   # hares -modify Hr_Mount01 FSType vxfs
   # hares -modify Hr_Mount01 FsckOpt %-n
   # hares -modify Hr_Mount01 MountOpt rw
   ```

   e. Add the Mount resource `Hr_Mount02` to mount the volume `hr_dv02` in the RVG resource `Hr_Rvg`:

   ```
   # hares -add Hr_Mount02 Mount ORAGrp
   # hares -modify Hr_Mount02 MountPoint /hr_mount02
   # hares -modify Hr_Mount02 BlockDevice \
      /dev/vx/dsk/Hr_Dg/hr_dv02
   # hares -modify Hr_Mount02 FSType vxfs
   # hares -modify Hr_Mount02 FsckOpt %-n
   # hares -modify Hr_Mount02 MountOpt rw
   ```

f. Add the Oracle resource `Hr_Oracle`

```
# hares -add Hr_Oracle Oracle ORAGrp
# hares -modify Hr_Oracle Sid hr1
# hares -modify Hr_Oracle Owner oracle
# hares -modify Hr_Oracle Home "/hr_mount01/OraHome1"
# hares -modify Hr_Oracle Pfile "inithr1.ora"
# hares -modify Hr_Oracle User dbtest
# hares -modify Hr_Oracle Pword dbtest
# hares -modify Hr_Oracle Table oratest
# hares -modify Hr_Oracle MonScript "./bin/Oracle/SqlTest.pl"
# hares -modify Hr_Oracle StartUpOpt STARTUP
# hares -modify Hr_Oracle ShutDownOpt IMMEDIATE
# hares -modify Hr_Oracle AutoEndBkup 1
```

g. Add the Oracle listener resource `LISTENER`

```
# hares -add LISTENER Netlsnr ORAGrp
# hares -modify LISTENER Owner oracle
# hares -modify LISTENER Home "/hr_mount01/OraHome1"
# hares -modify LISTENER Listener LISTENER
# hares -modify LISTENER EnvFile "/oracle/.profile"
# hares -modify LISTENER MonScript "./bin/Netlsnr/LsnrTest.pl"
```

h. Add the RVGPrimary resource `Hr_RvgPri`

```
# hares -add Hr_RvgPri RVGPrimary ORAGrp
# hares -modify Hr_RvgPri RvgResourceName Hr_Rvg
```

i. Specify resource dependencies for the resources you added in the previous steps:

```
# hares -link LISTENER Hr_Oracle
# hares -link LISTENER oraip
# hares -link Hr_Oracle Hr_Mount01
# hares -link Hr_Oracle Hr_Mount02
# hares -link Hr_Mount01 rvg-pri
# hares -link Hr_Mount02 rvg-pri
# hares -link oraip oranic
```

j. Specify an online local hard group dependency between `ORAGrp` and `VVRGrp`.

```
# hagrp -link ORAGrp VVRGrp online local hard
```

k. Enable all resources in `ORAGrp`

```
# hagrp -enableresources ORAGrp
```

l. Save and close VCS configuration

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

5. Repeat steps 1 to 4 on the system `london1` in the Secondary cluster `London` with the changes described below:

   a. Repeat steps 1 and 2.

   b. At step 3a, replace `seattle1` and `seattle2` with `london1` and `london2`, as follows:

      Add a service group, `VVRGrp`, to the cluster `London` and modify the attributes `SystemList` and `AutoStartList` of the service group to populate `SystemList` and `AutoStartList`:

```
# hagrp -add VVRGrp
# hagrp -modify VVRGrp SystemList london1 0 london2 1
# hagrp -modify VVRGrp AutoStartList london1 london2
```

   c. Repeat steps 3b, 3c, 3d.

   d. At step 3e, modify the Address attribute for the IP resource appropriately.

      Add the IP resource `vvrip` to the service group `VVRGrp` and modify the attributes of the resource:

```
# hares -add vvrip IP VVRGrp
# hares -modify vvrip Device eth3
# hares -modify vvrip Address 192.2.40.21
# hares -modify vvrip NetMask "255.255.248.0"
```

   e. Repeat steps 3f and 3g.

   f. At step 4a, replace `seattle1` and `seattle2` with `london1` and `london2`, as follows:

      Add a service group, `ORAGrp`, to the cluster `London` and populate the attributes `SystemList`, `AutoStartList` and `ClusterList` of the service group

```
# hagrp -add ORAGrp
# hagrp -modify ORAGrp SystemList london1 0 london2 1
# hagrp -modify ORAGrp AutoStartList london1 london2
# hagrp -modify ORAGrp ClusterList  Seattle 0 London 1
```

   g. Repeat step 4b

h. At step 4c, modify the Address attribute for the IP resource appropriately.

Add the IP resource `oraip` to the service group `ORAGrp` and modify the attributes of the resource:

```
# hares -add oraip IP ORAGrp
# hares -modify oraip Device eth0
# hares -modify oraip Address 192.2.40.1
# hares -modify oraip NetMask "255.255.248.0"
```

i. Repeat steps 4d through 4l.

6. Bring the service groups online, if not already online.

```
# hagrp -online VVRGrp -sys seattle1
# hagrp -online ORAGrp -sys seattle1
```

7. Verify that the service group `ORAGrp` is ONLINE on the system `seattle1` by issuing the following command:

```
# hagrp -state ORAGrp
```

Configuring the Agents When VCS is Stopped

Perform the following steps to configure the RVG agent using the sample configuration file on the first node in the Primary cluster and Secondary cluster. In the example in this guide, seattle1 is the first Primary node and london1 is the first Secondary node.

1. Log in as root.

2. Ensure that all changes to the existing configuration have been saved and that further changes are prevented while you modify `main.cf`:

If the VCS cluster is currently writeable, run the following command:

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

If the VCS cluster is already read only, run the following command:

```
# haconf -dump
```

3. Do not edit the configuration files while VCS is started. The following command will stop the `had` daemon on all systems and leave resources available:

```
# hastop -all -force
```

4. Make a backup copy of the main.cf file:

```
# cd /etc/VRTSvcs/conf/config
# cp main.cf main.cf.orig
```

5. Edit the `main.cf` files for the Primary and Secondary clusters. The files `main.cf.seattle` and `main.cf.london` located in the `/etc/VRTSvcs/conf/sample_vvr/RVGPrimary` directory can be used for reference for the primary cluster and the secondary cluster respectively.

6. Save and close the file.

7. Verify the syntax of the file `/etc/VRTSvcs/conf/config/main.cf`:

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

8. Start the VCS engine:

   ```
   # hastart
   ```

9. Go to "Administering the Service Groups" on page 32.

Administering the Service Groups

This section explains how to administer a VCS service group for cluster `Seattle` from the command line. Note that you can also use the VCS Java and Web consoles to administer service groups.

1. Start the VCS engine on `seattle1`:

   ```
   # hastart
   ```

2. Verify that all the service groups that contain RVG resource type are brought online:

   ```
   # hagrp -display
   ```

3. Take the service group offline and verify that all resources are stopped:

   ```
   # hagrp -offline hr_grp -sys seattle1
   # hagrp -display
   ```

4. Bring the service group online again and verify that all resources are available:

   ```
   # hagrp -online hr_grp -sys seattle1
   # hagrp -display
   ```

5. Start the VCS engine on `seattle2`:

   ```
   # hastart
   ```

6. Switch the VVR service group to `seattle2`:

   ```
   # hagrp -switch hr_grp -to seattle2
   ```

7. Verify that all the service groups that contain RVG resource type are brought online on `seattle2`:

   ```
   # hagrp -display
   ```

8. Repeat step 1 through step 7 for the cluster `London`.

9. If required, check the following log files on any system for the status or any errors:

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

   ```
   /var/VRTSvcs/log/RVG_A.log
   ```

Modifying the Agent Configuration

You can dynamically configure or modify the VCS agents for VVR from the command line or from the VCS Java and Web consoles. For instructions, see the chapters on administering VCS in the *VERITAS Cluster Server User's Guide.*

Index