

Veritas™ Cluster Server Agents for Veritas™ Volume Replicator Configuration Guide

HP-UX 11i v3

5.1 Service Pack 1



Veritas™ Cluster Server Agents for Veritas™ Volume Replicator Configuration Guide

The software described in this book is furnished under a license agreement and may be used only in accordance with the terms of the agreement.

Product version: 5.1 SP1

Document version: 5.1SP1.0

Legal Notice

Copyright © 2011 Symantec Corporation. All rights reserved.

Symantec, the Symantec logo, Veritas, Veritas Storage Foundation, CommandCentral, NetBackup, Enterprise Vault, and LiveUpdate are trademarks or registered trademarks of Symantec corporation or its affiliates in the U.S. and other countries. Other names may be trademarks of their respective owners.

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 Symantec Corporation 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. SYMANTEC CORPORATION 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, "Rights in Commercial Computer Software or Commercial Computer Software Documentation", as applicable, and any successor regulations. 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.

Symantec Corporation
350 Ellis Street
Mountain View, CA 94043
<http://www.symantec.com>

Technical Support

Symantec Technical Support maintains support centers globally. Technical Support's primary role is to respond to specific queries about product features and functionality. The Technical Support group also creates content for our online Knowledge Base. The Technical Support group works collaboratively with the other functional areas within Symantec to answer your questions in a timely fashion. For example, the Technical Support group works with Product Engineering and Symantec Security Response to provide alerting services and virus definition updates.

Symantec's support offerings include the following:

- A range of support options that give you the flexibility to select the right amount of service for any size organization
- Telephone and/or Web-based support that provides rapid response and up-to-the-minute information
- Upgrade assurance that delivers software upgrades
- Global support purchased on a regional business hours or 24 hours a day, 7 days a week basis
- Premium service offerings that include Account Management Services

For information about Symantec's support offerings, you can visit our Web site at the following URL:

www.symantec.com/business/support/index.jsp

All support services will be delivered in accordance with your support agreement and the then-current enterprise technical support policy.

Contacting Technical Support

Customers with a current support agreement may access Technical Support information at the following URL:

www.symantec.com/business/support/contact_techsupp_static.jsp

Before contacting Technical Support, make sure you have satisfied the system requirements that are listed in your product documentation. Also, you should be at the computer on which the problem occurred, in case it is necessary to replicate the problem.

When you contact Technical Support, please have the following information available:

- Product release level

- Hardware information
- Available memory, disk space, and NIC information
- Operating system
- Version and patch level
- Network topology
- Router, gateway, and IP address information
- Problem description:
 - Error messages and log files
 - Troubleshooting that was performed before contacting Symantec
 - Recent software configuration changes and network changes

Licensing and registration

If your Symantec product requires registration or a license key, access our technical support Web page at the following URL:

www.symantec.com/business/support/

Customer service

Customer service information is available at the following URL:

www.symantec.com/business/support/

Customer Service is available to assist with non-technical questions, such as the following types of issues:

- Questions regarding product licensing or serialization
- Product registration updates, such as address or name changes
- General product information (features, language availability, local dealers)
- Latest information about product updates and upgrades
- Information about upgrade assurance and support contracts
- Information about the Symantec Buying Programs
- Advice about Symantec's technical support options
- Nontechnical presales questions
- Issues that are related to CD-ROMs or manuals

Support agreement resources

If you want to contact Symantec regarding an existing support agreement, please contact the support agreement administration team for your region as follows:

Asia-Pacific and Japan	customercare_apac@symantec.com
Europe, Middle-East, and Africa	semea@symantec.com
North America and Latin America	supportsolutions@symantec.com

Documentation

Product guides are available on the media in PDF format. Make sure that you are using the current version of the documentation. The document version appears on page 2 of each guide. The latest product documentation is available on the Symantec Web site.

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

Your feedback on product documentation is important to us. Send suggestions for improvements and reports on errors or omissions. Include the title and document version (located on the second page), and chapter and section titles of the text on which you are reporting. Send feedback to:

docs@symantec.com

About Symantec Connect

Symantec Connect is the peer-to-peer technical community site for Symantec's enterprise customers. Participants can connect and share information with other product users, including creating forum posts, articles, videos, downloads, blogs and suggesting ideas, as well as interact with Symantec product teams and Technical Support. Content is rated by the community, and members receive reward points for their contributions.

<http://www.symantec.com/connect/storage-management>

Contents

Technical Support	4	
Chapter 1	Overview of the VCS agents for VVR	9
	Introducing the VCS agents for VVR	9
	How the agents for failover applications work	10
	RVG agent	10
	RVGPrimary agent	13
	RVGSnapshot	19
	How the agents for parallel applications work	22
	RVGShared agent	22
	RVGLogowner agent	25
	RVGSharedPri agent	28
	How the agents for hybrid applications work	31
	Overview of how to configure VVR in a VCS environment	32
	Overview of how to configure the primary-elect feature in a VCS globally clustered environment	32
	Generic VVR setup in a VCS environment	33
	Example VVR configuration in a VCS environment	34
Chapter 2	Configuring the agents for high availability	35
	Requirements for configuring VVR in a VCS environment	35
	Best practices for setting up the agents	36
	Example configuration for a failover application	37
	Example configuration for a parallel application	38
	Example—setting up VVR in a VCS environment	40
	Setting up the VVR configuration	41
	Verifying the VVR replication state	44
	Configuring the agents for failover applications	44
	Configuring the agents for parallel applications	52
	Configuring the agents for a bunker replication configuration	54
	VCS configuration for a bunker using the STORAGE protocol	54
	VCS configuration for a bunker using IP	57
	How the RVGPrimary works in a bunker setup	57
	Administering the service groups	59

Appendix A	Sample main.cf files	61
	Globally clustered VCS and VVR main.cf	61
Index		67

Overview of the VCS agents for VVR

This chapter includes the following topics:

- [Introducing the VCS agents for VVR](#)
- [How the agents for failover applications work](#)
- [How the agents for parallel applications work](#)
- [How the agents for hybrid applications work](#)
- [Overview of how to configure VVR in a VCS environment](#)
- [Overview of how to configure the primary-elect feature in a VCS globally clustered environment](#)
- [Generic VVR setup in a VCS environment](#)
- [Example VVR configuration in a VCS environment](#)

Introducing the VCS agents for VVR

VCS provides agents that manage applications and resources in a cluster.

The different types of agents follow:

- VCS comes packaged (bundled) with a set of agents that enable VCS to provide high availability. These include agents for mount points, IP addresses, file systems, VVR, and virtual environments. These agents are immediately available to you after installing VCS.

For more information about VCS bundled agents, refer to the *Veritas Cluster Server One Bundled Agents Reference Guide*.

Agents are processes that manage predefined resource types. When an agent is started, it obtains configuration information from 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

How the agents for failover applications work

The agents for failover applications include the following:

- See [“RVG agent”](#) on page 10.
- See [“RVGPrimary agent”](#) on page 13.
- See [“RVGSnapshot”](#) on page 19.

RVG agent

Brings the RVG online, monitors read and write access to the RVG, and takes the RVG offline. This is a failover resource. The RVG agent enables replication between clusters. It manages the Primary VVR node in one cluster and the Secondary VVR node in another cluster. Each node can be failed over in its respective cluster. In this way, replication is made highly available.

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.

Using a VCS global cluster 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.
- Enables you to bring a service group online to start VCS-managed applications that use VVR.
- 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: This release does not support the attributes Primary, SRL, and RLinks of the RVG agent. If you have a configuration from a previous release, you must remove these attributes during the upgrade or the configuration will fail.

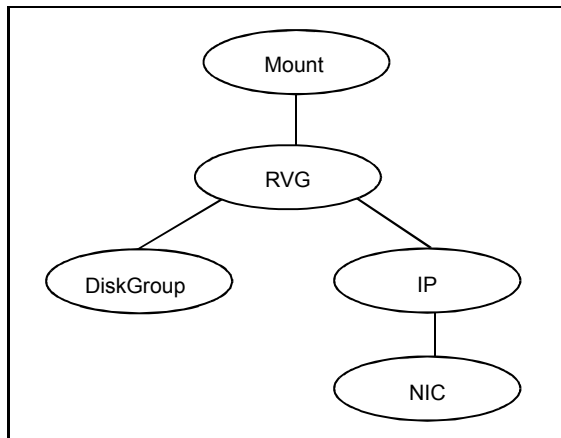
Resource dependencies for the RVG agent

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 Administrator's Guide* for more information on dependencies.

Figure 1-1 Sample service group for an RVG resource



RVG agent functions

The RVG agent has the following agent functions:

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.
Monitor	Monitors the state of the RVG using the <code>vxprint</code> command. The RVG resource monitors an RVG for local access only. It does not monitor replication.
Clean	Stops the RVG.
Info	Gives the information about the replication status for the Replicated Data Set (RDS). 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: <pre># hatype -modify resourcetype_name InfoInterval interval</pre> For example, to set the info interval to 60 seconds for the RVG resource type, enter: <pre># hatype -modify RVG InfoInterval 60</pre> The info interval indicates how frequently VCS executes the info entry point to update 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: <pre># hares -value resource_name ResourceInfo</pre> The output of the info entry point is also logged in the file <code>/var/VRTSvcs/log/engine_A.log</code> .

State definitions for the RVG agent

The RVG agent has the following state definitions:

ONLINE	Indicates that the RVG is in <code>ENABLED/ACTIVE</code> state.
OFFLINE	Indicates that the RVG is in <code>DISABLED/CLEAN</code> state.
FAULTED	The RVG resource fails if the RVG is not in the <code>ENABLED/ACTIVE</code> state.

Attribute definitions for the RVG agent

Table 1-1 The required attributes for the RVG agent

Attribute	Description
RVG	The name of the RVG being monitored. Type and dimension: string-scalar
DiskGroup	The disk group that this RVG is associated with. Type and dimension: string-scalar
StorageDG	The name of the bunker disk group. Type and dimension: string-scalar
StorageRVG	The name of the bunker RVG. Type and dimension: string-scalar
StorageHostIds	A space-separated list of the hostids of each node in the bunker cluster. Type and dimension: string-keylist

Resource type definitions for the RVG agent

The RVG agent resource type definition follows.

```
type RVG (
    static int NumThreads = 1
    static str ArgList[] = { RVG, DiskGroup }
    str RVG
    str DiskGroup
    str StorageRVG
    str StorageDG
    str StorageHostIds
)
```

RVGPrimary agent

Attempts to migrate or takeover a Secondary to a Primary upon an application failover. 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.

The agent is useful when hosts in both the primary and secondary side are clustered, in particular a VCS replicated data cluster or a VCS global cluster, to

completely automate the availability of writable replicated disks to a VCS-managed application.

The RVGPrimary agent includes the following features:

- Removes the 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.
- Allows you to distinguish the Primary site after network failure or disaster

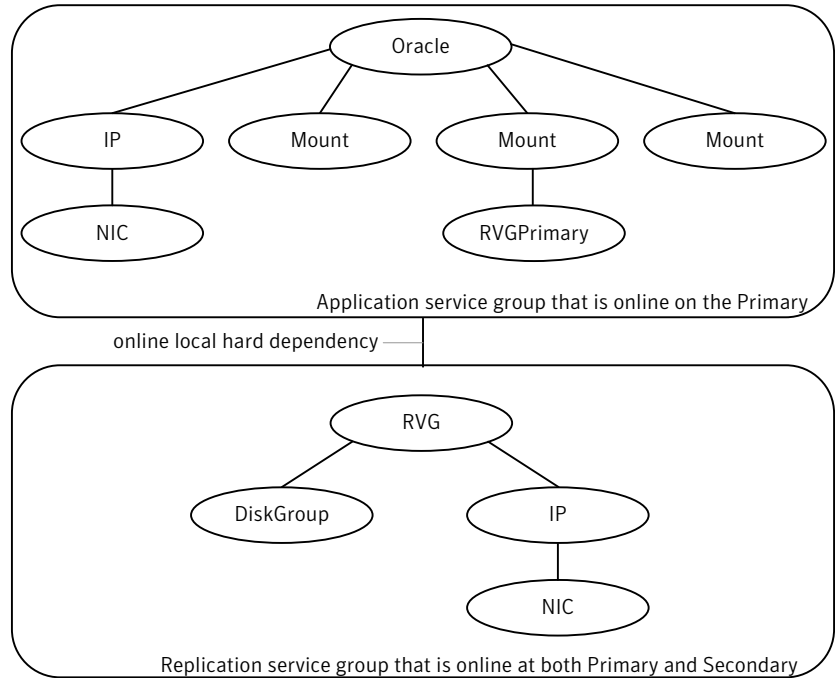
A sample configuration file for this agent that you can use as a guide to create the configuration is located at `/etc/VRTSvcs/conf/sample_vvr/RVGPrimary`.

Resource dependencies for the RVGPrimary agent

You usually use the RVGPrimary agent in conjunction with the RVG agent in two groups with an online local hard group dependency. The parent group contains the resources that manage the actual application and file systems and as the RVGPrimary resource. The child group contains the resources managing the storage infrastructure, which include the RVG and DiskGroup type resources.

Refer to the *Veritas Cluster Server Administrator's Guide* for information about the setup of a VVR environment using the RVGPrimary agent.

Figure 1-2 Sample service group for an RVGPrimary resource



RVGPrimary agent functions

The RVGPrimary agent has the following agent functions:

Online	Determines the current role of the RVG. If the role is Secondary it attempts a migration. It waits for any outstanding writes from the original Primary. If the original Primary is down, it attempts a takeover. If the RVG is a Primary, it performs no actions and goes online.
Offline	Performs no actions.
Monitor	Performs no actions. The RVG agents monitors the actual RVG.
Clean	Performs no actions.

State definitions for the RVGPrimary agent

The RVGPrimary agent has the following state definitions:

ONLINE	Indicates that the role of the RVG is Primary.
--------	--

FAULTED The RVG agents monitors the actual RVG. Accidental migration of a VVR Primary outside of VCS causes other resources to fault immediately, such as Mount. No special monitoring by this agent is necessary.

Attribute definitions for the RVGPrimary agent

Table 1-2 The required attributes for the RVGPrimary agent

Attribute	Description
RvgResourceName	The name of the RVG resource type that this agent promotes. The name RVG resource type which has been configured using the RVG agent. Type and dimension: string-scalar
AutoTakeover	A flag to indicate whether the agent should perform a takeover on online if the original Primary is down. AutoTakeover and AutoResync are mutually exclusive attributes. Type and dimension: integer-scalar
AutoResync	Indicates 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. You can use the following values for this attribute: <ul style="list-style-type: none"> ■ 0—instructs the agent to not attempt to perform a fast-failback resynchronization of the original Primary after a takeover and after the original Primary returns. ■ 1—instructs the agent to attempt to automatically perform a fast-failback resynchronization of the original Primary after a takeover and after the original Primary returns. AutoTakeover and AutoResync are mutually exclusive attributes. Type and dimension: integer-scalar

Resource type definitions for the RVGPrimary agent

The RVGPrimary resource type definition follows.

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



```
    str RvgResourceName
    int AutoTakeover = 1
    int AutoResync = 0
    str BunkerSyncTimeOut
    int BunkerSyncElapsedTime = 0
)
```

The resource type definition represents the VCS One configuration of the agent and specifies how the agent is defined in the main.xml configuration file. For more information, refer to the RVGPrimary type definition in types.linuxs390x.xml file in /etc/VRTSvcsone/conf/confxml directory on the Policy Master system.

Using the RVGPrimary agent for migration and takeover

The RVGPrimary agent can now handle the migration and takeover when you have multiple secondary sites that you have configured for VCS disaster recovery and global failover (n-way GCO).

In the case of a takeover, after the first failover succeeds, the rlinks from remaining secondaries to the new primary need to be attached. Attach these rlinks before trying subsequent failovers and to be ready for the next failovers to succeed. (Note that this is not the case for migration.)

If a bunker site is associated with the new primary, the agent tries to attach the bunker's rlink after a successful migration.

Consider an example of a disaster recovery (DR) setup where two DR sites are configured as follows:

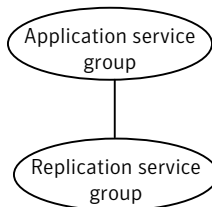
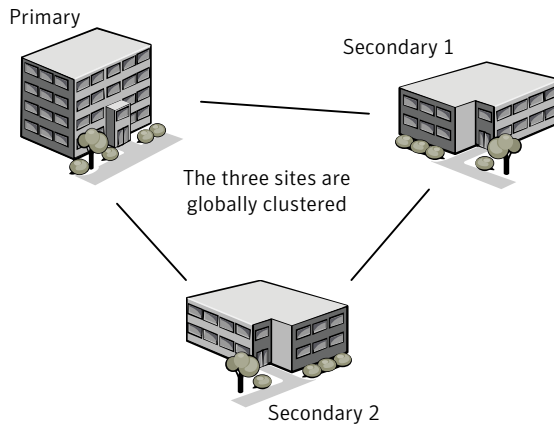
- Primary RVG: PRI_RVG
 - PRI_RVG has the following rlinks:
 - rlk_SEC_RVG1
 - rlk_SEC_RVG2
- Secondary RVG: SEC_RVG1
 - SEC_RVG1 has the following rlinks:
 - rlk_PRI_RVG
 - rlk_SEC_RVG2
- Secondary RVG: SEC_RVG2
 - SEC_RVG2 has the following rlinks:
 - rlk_PRI_RVG
 - rlk_SEC_RVG1

When the Primary site is failed over to secondary site, SEC_RVG1 becomes new primary.

Use the `vradmin` or `vxrlink` commands to attach the following links:

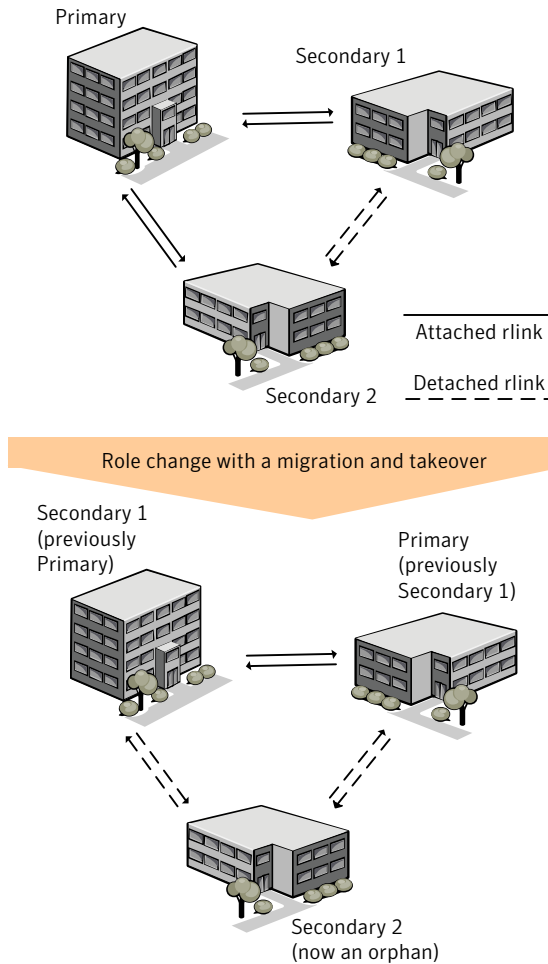
- `rlk_SEC_RVG2` that is associated with SEC_RVG1.
- `rlk_PRI_RVG` that is associated with SEC_RVG2.

Figure 1-3 RVGPrimary three-way global clustering support



The application's service group depends on the replication's service group

Figure 1-4 RVGPrimary three-way global clustering migration and takeover



RVGSnapshot

Creates and destroys a transactionally consistent space-optimized snapshot of all volumes in a VVR secondary replicated data set. The RVGSnapshot agent takes space-optimized snapshots on a secondary RVG. These snapshots can be mounted and written to without affecting the actual replicated data, which means that the 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 the VCS Mount agent, the CFS CFMount agent, and VCS agents that manage the

application being replicated, you can create a special fire drill service group. You can bring this service group online and take it offline at regularly scheduled intervals to verify that the disaster recovery environment is robust.

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 Administrator's Guide*.

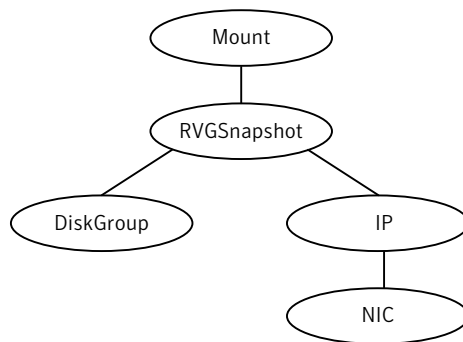
The RVGSnapshot agent includes the following key features:

- 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.

Resource dependencies for the RVGSnapshot agent

The RVGSnapshot agent depends on these resources.

Figure 1-5 Sample service group for an RVGSnapshot resource



RVGSnapshot agent functions

The RVGSnapshot agent has the following agent functions:

- | | |
|---------|---|
| Online | Creates a transactionally consistent snapshot of all volumes in the RDS. |
| Offline | Destroys the snapshot. |
| Monitor | No operation; failure of the snapshot will be indicated by the failure of the Mount resource of any file systems mounted on it. |

Clean Cleans up any failed snapshot creation or deletion.

State definitions for the RVGSnapshot agent

The RVGSnapshot agent has the following state definitions:

ONLINE Indicates that a snapshot was created.

OFFLINE Indicates that a snapshot was destroyed.

FAULTED The RVGSnapshot resource faults on timeout if a snapshot creation did not succeed during an online.

Attribute definitions for the RVGSnapshot agent

Table 1-3 The required attributes for the RVGSnapshot agent

Attribute	Description
RvgResourceName	The name of the VCS RVG-type resource that manages the RVG that will be snapshot by this agent. Type and dimension: string-scalar
CacheObj	Name of the cache object that is required for a space-optimized snapshot; the fdsetup wizard will create one if it does not exist Type and dimension: string-scalar
Prefix	Token put before the name of the actual volume when creating the snapshotted volumes. Type and dimension: string-scalar

Table 1-4 The optional attributes for the RVGSnapshot agent

Attribute	Description
DestroyOnOffline	A flag to indicate whether to destroy the snapshot upon taking the resources offline. 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). Type and dimension: integer-scalar Default: 1

Table 1-4 The optional attributes for the RVGSnapshot agent (*continued*)

Attribute	Description
FDFile	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 and dimension: string-scalar

Resource type definitions for the RVGSnapshot agent

The resource type definition for the RVGSnapshot agent follows.

```
type RVGSnapshot (  
    static keylist RegList = { Prefix }  
    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 parallel applications work

The agents for parallel applications include the following:

- See [“RVGShared agent”](#) on page 22.
- See [“RVGLogowner agent”](#) on page 25.
- See [“RVGSharedPri agent”](#) on page 28.
- See [“RVGSnapshot”](#) on page 19.

RVGShared agent

Monitors the RVG in a shared environment. This is a parallel resource. The RVGShared agent enables you to configure parallel applications to use an RVG in a cluster. The RVGShared agent monitors the RVG in a shared disk group environment. The RVGShared agent must be configured as a parallel group in VCS. Typically, the RVGShared resource is online or offline at the same time on all the nodes in the VCS cluster. 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/RVGLogowner`.

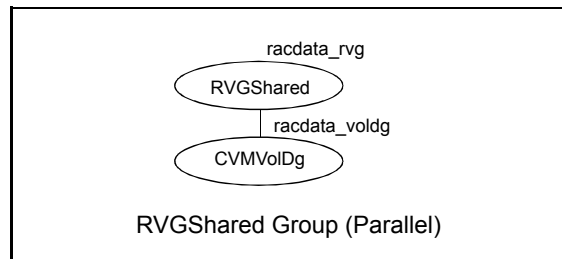
Resource dependencies for the RVGShared agent

The RVGShared resource represents the RVG of the RDS. The RVGShared resource is dependent on the CVMVolDg resource.

The RVGShared resource must be configured in a parallel group.

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

Figure 1-6 Sample service group for an RVGShared resource



Note: Do not add any volumes that are part of the RVG in the CVMVolume attribute of the CVMVolDg resource. The volumes in the RVG are managed by the RVGShared resource.

RVGShared agent functions

The RVGShared agent has the following agent functions:

Online	Verifies whether the RVG is started. If the RVG is not started, recovers and starts the RVG.
Offline	No action.
Monitor	Displays the state as <code>ONLINE</code> if the RVG is started. Displays the state as <code>OFFLINE</code> if the RVG is not started.
Clean	No action.

Info Gives the information about the replication status for the Replicated Data Set (RDS).

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 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`.

State definitions for the RVGShared agent

The RVGShared agent has the following state definitions:

- ONLINE** Indicates that the RVG is in the `ENABLED/ACTIVE` state.
- OFFLINE** Indicates that the RVG is not in the `ENABLED/ACTIVE` state or that the administrator has invoked the offline entry point.

Attribute definitions for the RVGShared agent

Table 1-5 The required attributes for the RVGShared agent

Attribute	Description
RVG	The name of the RVG being monitored. Type and dimension: string-scalar
DiskGroup	The shared-disk group with which this RVG is associated. Type and dimension: string-scalar

Resource type definitions for the RVGShared agent

The RVGShared resource type definition follows.

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

RVGLogowner agent

Assigns and unassigns a node as the logowner in the CVM cluster; this is a failover resource. The RVGLogowner agent assigns or unassigns a node as a logowner in the cluster. To replicate data, VVR requires network connectivity between the Primary and the Secondary. In a shared disk group environment, only one node, that is, the logowner, can replicate data to the Secondary.

For replication to be highly available, the logowner must be highly available. To make the logowner highly available, the RVGLogowner resource must be configured as a resource in a failover group. Also, a virtual IP must be set up on the logowner to enable replication and failover of the logowner from one node to another in a cluster. The virtual IP must be configured as an IP resource.

For more information about the logowner, see the *Veritas Volume Replicator Administrator's Guide*. 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/RVGLogowner`.

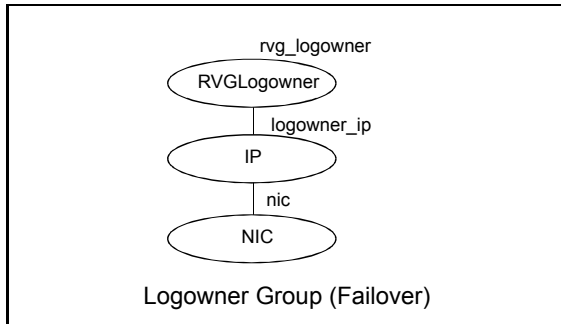
Resource dependencies for the RVGLogowner agent

The RVGLogowner resource represents the logowner for RVG in the cluster. The RVGLogowner resource is dependent on the IP resource that it uses for replication.

The RVGLogowner resource must be configured in a failover group. The RVGLogowner group is used in conjunction with the RVGSharedPri and RVGShared agents in separate groups, with the appropriate service group dependencies.

For more information on dependencies, refer to the *Veritas Cluster Server Administrator's Guide*

Figure 1-7 Sample service group for an RVGLogowner resource



RVGLogowner agent functions

The RVGLogowner agent has the following agent functions:

Online	Assigns the logowner on the node.
Offline	Unassigns the logowner on the node.
Monitor	Returns <code>ONLINE</code> if the node is the logowner and the RVG is in <code>ENABLED/ACTIVE</code> state. Returns <code>OFFLINE</code> if the node is the logowner and the state is not <code>ENABLED/ACTIVE</code> , or if the node is not the logowner (regardless of the state). The RVG for which the logowner is monitored must be configured as the <code>RVGShared</code> resource type.
Clean	Unassigns the logowner on the node.

State definitions for the RVGLogowner agent

The RVGLogowner agent has the following state definitions:

<code>ONLINE</code>	Indicates that the node is the logowner for the RVG in the cluster.
<code>OFFLINE</code>	Indicates that the node is not the logowner for the RVG in the cluster.

Attribute definitions for the RVGLogowner agent

Table 1-6 The required attributes for the RVGLogowner agent

Attribute	Description
RVG	The name of the RVG being monitored. Type and dimension: string-scalar

Table 1-6 The required attributes for the RVGLogowner agent (*continued*)

Attribute	Description
DiskGroup	The disk group with which this RVG is associated. Type and dimension: string-scalar

Table 1-7 The bunker attributes for the RVGLogowner agent

Attribute	Description
StorageDG	The name of the bunker disk group. Type and dimension: string-scalar
StorageRVG	The name of the bunker RVG. Type and dimension: string-scalar
StorageHostIds	A space-separated list of the hostids of each node in the bunker cluster. Type and dimension: string-keylist

Resource type definitions for the RVGLogowner agent

The RVGLogowner resource type definition follows.

```
type RVGLogowner (
    static int NumThreads = 1
    static str ArgList[] = { RVG, DiskGroup }
    static int OnlineRetryLimit = 5
    str RVG
    str DiskGroup
    str StorageRVG
    str StorageDG
    str StorageHostIds
)
```

Notes for the RVGLogowner agent

Review the following notes for more information on the RVGLogowner agent.

CVM master node needs to assume the logowner role for VCS managed VVR resources

If you use VCS to manage RVGLogowner resources in an SFCFS environment or an SF Oracle RAC environment, Symantec recommends that you perform the

following procedures. These procedures ensure that the CVM master node always assumes the logowner role. Not performing these procedures can result in unexpected issues that are due to a CVM slave node that assumes the logowner role.

For a service group that contains an RVGLogowner resource, change the value of its PreOnline trigger to 1 to enable it.

To enable the PreOnline trigger from the command line on a service group that has an RVGLogowner resource

- ◆ On each node in the cluster, perform the following command:

```
# hagrps -modify RVGLogowner_resource_sg PreOnline 1 -sys system
```

Where *RVGLogowner_resource_sg* is the service group that contains the RVGLogowner resource. The *system* is the name of the node where you want to enable the trigger.

On each node in the cluster, merge the `preonline_vvr` trigger into the default `triggers` directory.

To merge the `preonline_vvr` trigger

- ◆ On each node in the cluster, merge the `preonline_vvr` trigger to the `/opt/VRTSvcs/bin/triggers` directory.

```
# cp /opt/VRTSvcs/bin/sample_triggers/preonline_vvr \  
/opt/VRTSvcs/bin/triggers
```

Refer to the sample configurations directory for samples of how to enable these triggers (`/opt/VRTSvcs/bin/sample_triggers`).

RVGSharedPri agent

Attempts to migrate or takeover a Secondary to a Primary when a parallel service group fails over. The RVGSharedPri agent enables migration and takeover of a VVR replicated data set in parallel groups in a VCS environment. Bringing a resource of type RVGSharedPri online causes the RVG on the local host to become a primary if it is not already. The agent is useful when hosts in both the primary and secondary side are clustered using a VCS global cluster, to completely automate the availability of writable replicated disks to an application managed by VCS.

You cannot use the primary-elect feature with this agent. For a detailed description of the primary-elect feature, see *Veritas Volume Replicator Administrator's Guide*.

The RVGSharedPri 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.

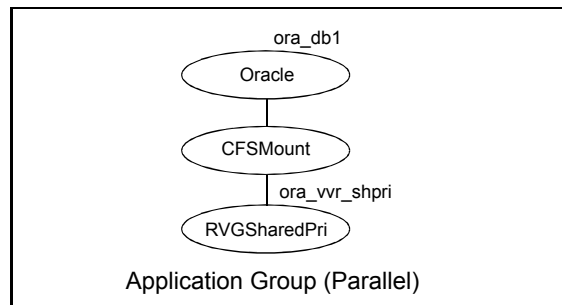
Sample configuration files are located in the `/etc/VRTSvcs/conf/sample_rac/` directory and include CVR in the filename. These sample files are installed as part of the VRTSdbac package, and can be used as a guide when creating your configuration.

Resource dependencies for the RVGSharedPri agent

The RVGSharedPri agent is used in conjunction with the RVGShared and RVGLogowner agents in separate groups, with the appropriate service group dependencies.

The RVGSharedPri agent must be configured in a parallel service group. The application service group contains the resources managing the actual application and file systems as well as the RVGSharedPri agent.

Figure 1-8 Sample service group for an RVGSharedPri resource



RVGSharedPri agent functions

The RVGSharedPri agent has the following agent functions:

- | | |
|---------|--|
| 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 | Performs no actions. |

- Monitor Performs no actions; monitoring of the actual RVG is done by the RVGShared agent.
- Clean Performs no actions.

State definitions for the RVGSharedPri agent

The RVGSharedPri agent has the following state definitions:

- FAULTED** Monitoring of the actual RVG is done by the RVGShared 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.

Attribute definitions for the RVGSharedPri agent

Table 1-8 The required attributes for the RVGSharedPri agent

Attribute	Description
RvgResourceName	The name of the RVGShared resource type that this agent will promote, that is, the name RVG resource type which has been configured using the RVGShared agent. The required VVR object names, such as the name of the RVG, Disk Group, RLINKs, SRL are discovered by this agent by querying VCS directly. Type and dimension: string-scalar
AutoTakeover	A flag to indicate whether the agent should perform a takeover on online if the original Primary is down. AutoTakeover and AutoResync are mutually exclusive attributes. Type and dimension: integer-scalar Default: 1
AutoResync	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. AutoTakeover and AutoResync are mutually exclusive attributes. Type and dimension: integer-scalar Default: 0
VCSResLock	This attribute is reserved for internal use by VCS. Type and dimension: string-scalar

Resource type definitions for the RVGSharedPri agent

The RVGSharedPri resource type definition follows.

```
type RVGSharedPri (
    static keylist SupportedActions = { fbsync, resync }
    static int NumThreads = 1
    static int OnlineRetryLimit = 1
    static str ArgList[] = { RvgResourceName, "RvgResourceName:RVG",
        "RvgResourceName:DiskGroup", AutoTakeover, AutoResync }
    str RvgResourceName
    int AutoTakeover = 1
    int AutoResync = 0
    temp str VCSResLock
)
```

How the agents for hybrid applications work

The agents for hybrid applications include the following:

- See [“RVG agent”](#) on page 10.
- See [“RVGPrimary agent”](#) on page 13.

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 See [“RVG agent”](#) on page 10. and See [“RVGPrimary agent”](#) on page 13.. These sections 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:

Table 1-9 Attribute for 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.

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 in the following order:

- Set up a VVR configuration, which involves creating a Replicated Data Set (RDS).
- Create service groups for the VVR agents and add the resource and group dependencies appropriately.
- Add these application service groups to the composite service group for global failover.

Overview of how to configure the primary-elect feature in a VCS globally clustered environment

Before Release 5.1 SP1, if a disaster struck at the Primary site or a network disruption, the applications were taken offline on the original Primary and failed over to the Secondary.

When the original Primary returned or the network disruption was corrected, you had the following options:

- Manually resynchronize the original Primary
- Automatically resynchronize the original Primary

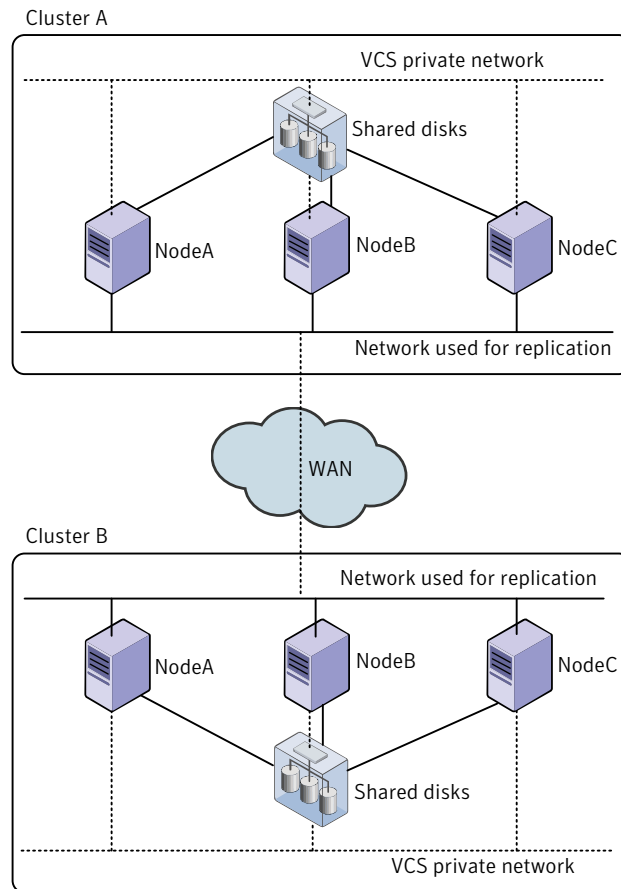
Beginning in Release 5.1 SP1, you have a third option. After the original Primary returns or the network disruption is corrected, you can specify which site is the Primary going forward. This feature is called the primary-elect feature, and it is enabled through the VCS Global Cluster Option. The key difference between the primary-elect feature and the other options is that if a network disruption occurs, applications continue to run on the Primary site and they are also failed over to

the Secondary. This allows you to maintain application availability on both sites while the network is down.

For a detailed description of the primary-elect feature, see *Veritas Volume Replicator Administrator's Guide*.

Generic VVR setup in a VCS environment

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

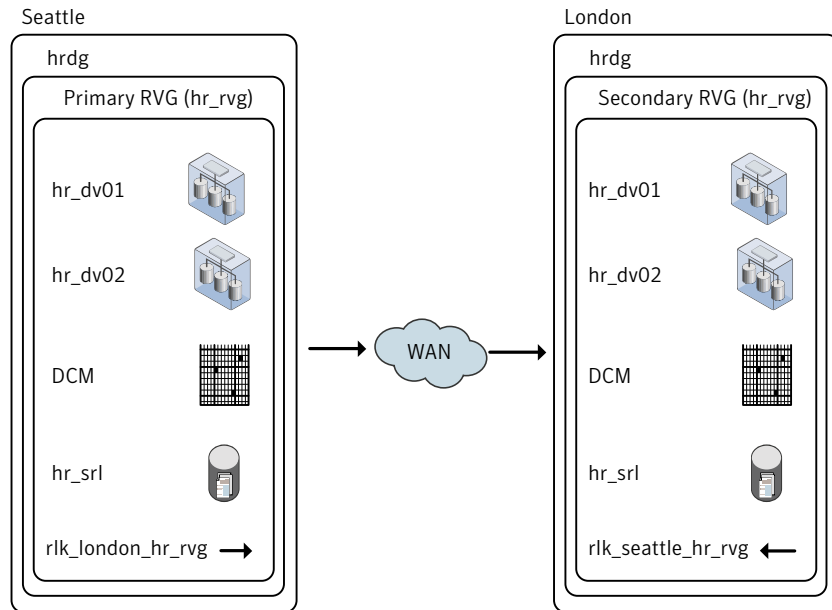


Example VVR configuration in a VCS environment

In the following example, two clusters reside at separate sites. VVR uses a WAN to replicate data between the sites.

The first cluster (Seattle) is in Seattle. The Seattle cluster consists of two nodes: seattle1 and seattle2. The second cluster is in London and is named London. The London cluster 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.

Figure 1-9 Example—VVR configuration in a VCS environment



Configuring the agents for high availability

This chapter includes the following topics:

- [Requirements for configuring VVR in a VCS environment](#)
- [Example configuration for a failover application](#)
- [Example configuration for a parallel application](#)
- [Example—setting up VVR in a VCS environment](#)
- [Configuring the agents for a bunker replication configuration](#)
- [Administering the service groups](#)

Requirements for configuring VVR in a VCS environment

The requirements for configuring VVR in a VCS environment are as follows:

- Follow the best practices for setting up replication with VVR.
For information about setting up replication, refer to the *Veritas Volume Replicator Administrator's Guide*.
- Each node that is part of a particular 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 the RVG Agent. VCS requires the `noautoimport` attribute of the disk group to be set.
Refer to the *Veritas Cluster Server Bundled Agents Reference Guide* for more information about setting the `noautoimport` attribute.

Best practices for setting up the agents

The following list gives the 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.
If a disk group is configured as a `CVMVolDg` resource, then all the `RVGs` must be configured as `RVGShared` resources.
- When configuring failover applications, use the `RVG` and `RVGPrimary` agents.
- When configuring parallel applications, use the `RVGShared`, `RVGSharedPri`, and `RVGLogowner` agents. If the configuration has multiple `RVGLogowner` resources, we recommend that you alternate the order of hosts in the `AutoStartList` attributes for the service groups containing the `RVGLogowner` resources. VCS then brings online the `RVGLogowner` resources on different nodes in the cluster, which facilitates load-balancing. For example, the first service group containing an `RVGLogowner` resource would appear as:

```
AutoStartList = { seattle1, seattle2 }
```

whereas the next service group would have:

```
AutoStartList = { seattle2, seattle1 } and so on.
```

- Do not configure the `RVGShared` resource in the `cvm` group. Configure the `RVGShared` resource in a separate group which contains the `RVGShared` resource and the `CVMVolDg` resource.
- In a global clustering environment, you must also use the `ClusterList` attribute.

```
ClusterList = { gco_primclus=1, gco_secclus=2 }
```
- If a volume set is fully associated to an `RVG`, that is, if all its component volumes are associated to the `RVG`, you can add the volume set to the agent configuration in the same way that a volume is added. Specify the volume set in the `Mount` resource instead of the component volume names.
See [“Example—setting up VVR in a VCS environment”](#) on page 40.

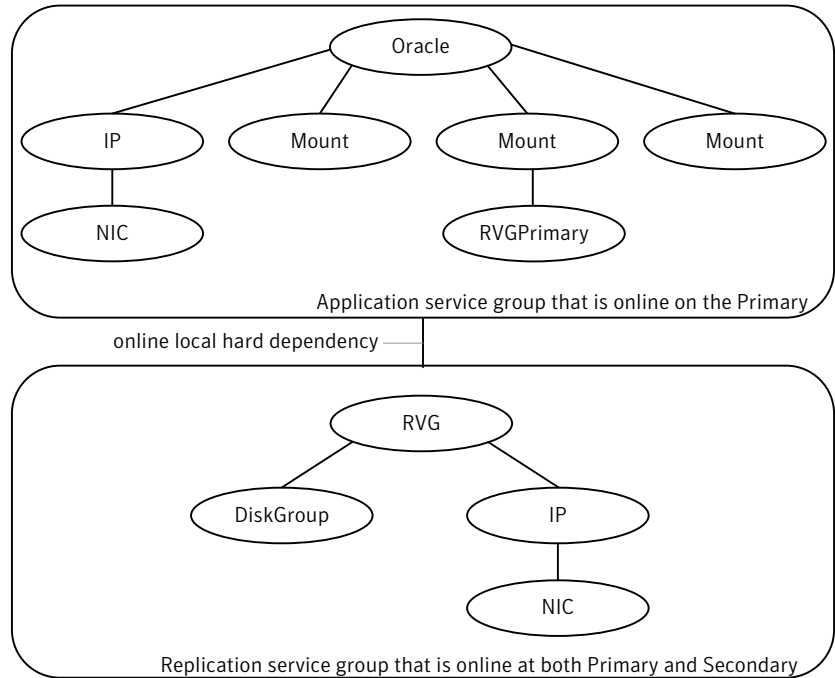
Note: The agents do not support mounting a volume set that is partially associated to an RVG, that is, if one or more of its component volumes are not associated to the RVG.

For more information about using volume sets in an RVG, refer to the *Veritas Volume Replicator Administrator's Guide*.

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.

Figure 2-1 RVG and RVGPrimary agents—service groups and resource dependencies

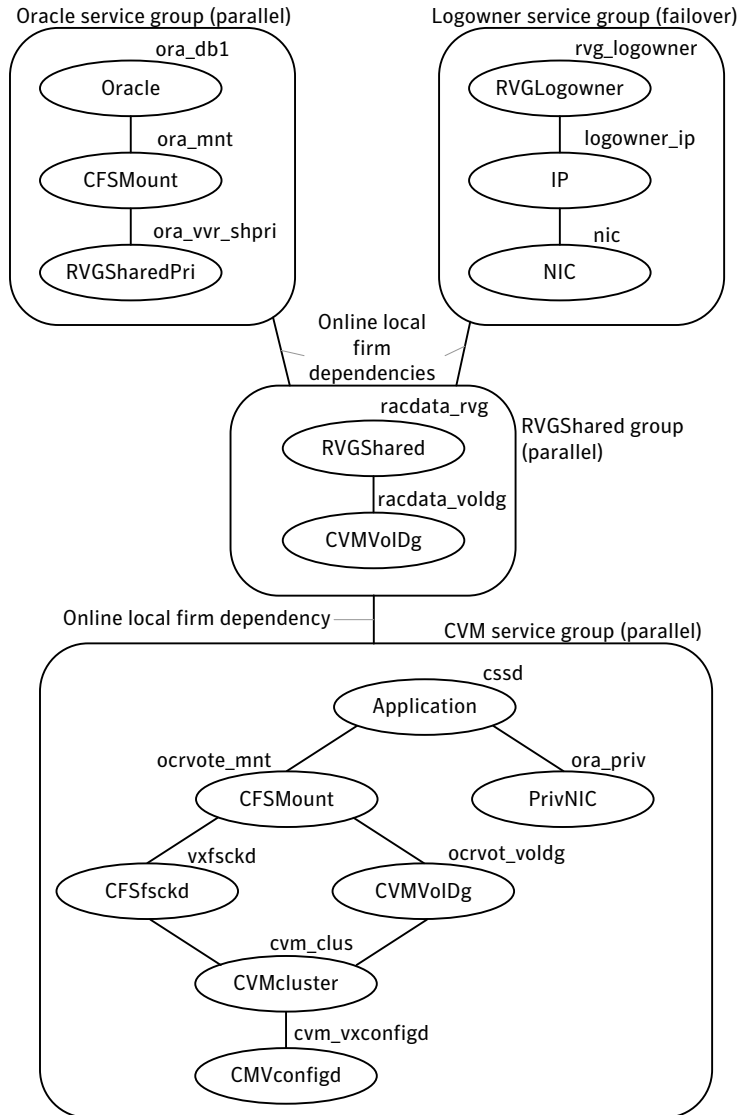


Example configuration for a parallel application

In the following example, a parallel application that uses an RVG is made highly available across two clusters. The Oracle service group is the application group and contains the CFSSMount resource. The Logowner service group is a failover group, which manages the logowner. The service group RVGShared manages the RVG used by the application. The Oracle and CVM groups are configured as parallel groups.

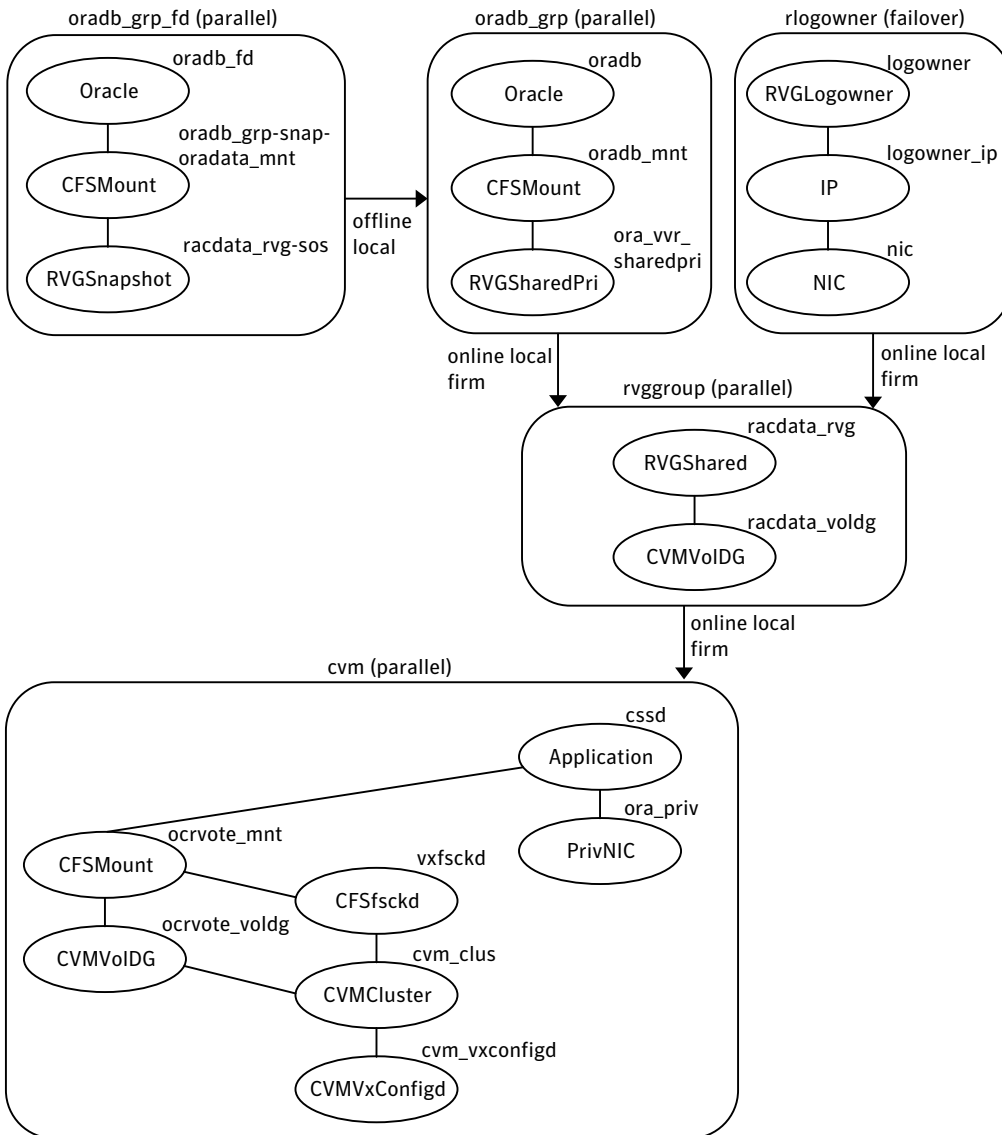
The service groups Logowner and Oracle are dependent on the service group RVGShared. The RVGShared manages the RVG in a shared environment; therefore, it is dependent on the cvm service group.

Figure 2-2 RVGShared, RVGLogowner, and RVGSharedPri agents—Service Groups and Resource Dependencies



The following diagram shows the service group dependencies for a sample FireDrill service group.

Figure 2-3 Example configuration of FireDrill for parallel application



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 following order:

- See “Setting up the VVR configuration” on page 41.
- See “Verifying the VVR replication state” on page 44.
- See “Configuring the agents for failover applications” on page 44.
- See “Configuring the agents for parallel applications” on page 52.

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. If the port number is the same on all nodes, add the VVR agents to the VCS configuration.

For instructions on using the `vrport` command, see the *Veritas Volume Replicator Administrator's Guide*.

Setting up the VVR configuration

This section shows how to set up a sample VVR configuration. The VVR configuration in this example applies to the RVG Agent.

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` 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 volume set (with data volumes hr_dv03, hr_dv04)	hr_vset01
Primary SRL for hr_rvg	hr_srl
Cluster IP address	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 volume set (with data volumes hr_dv03, hr_dv04)	hr_vset01
Secondary SRL for hr_rvg	hr_srl
Cluster IP address	10.216.144.162

In this example, each of the hosts (seattle1 and london1) has a disk group named hrdg with enough free space to create the VVR objects.

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.

See [“Example VVR configuration in a VCS environment”](#) on page 34.

To set up the VVR configuration

1 On london1:

- Create the Secondary data volumes.

```
# vxassist -g hrdg make hr_dv01 100M \  
    layout=mirror logtype=dcn mirror=2  
# vxassist -g hrdg make hr_dv02 100M \  
    layout=mirror logtype=dcn mirror=2
```

- Create the data volumes for the volume set on the Secondary and create the volume set.

```
# vxassist -g hrdg make hr_dv03 100M \  
    layout=mirror logtype=dcn mirror=2  
# vxassist -g hrdg make hr_dv04 100M \  
    layout=mirror logtype=dcn mirror=2  
# vxmake -g hrdg vset hr_vset01 \  
    appvols=hr_dv03,hr_dv04
```

- Create the Secondary SRL.

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

2 On seattle1:

- Create the Primary data volumes.

```
# vxassist -g hrdg make hr_dv01 100M \
    layout=mirror logtype=dcn mirror=2
# vxassist -g hrdg make hr_dv02 100M \
    layout=mirror logtype=dcn mirror=2
```

- Create the data volumes for the volume set on the Primary and create the volume set.

```
# vxassist -g hrdg make hr_dv03 100M \
    layout=mirror logtype=dcn mirror=2
# vxassist -g hrdg make hr_dv04 100M \
    layout=mirror logtype=dcn mirror=2
# vxmake -g hrdg vset hr_vset01 \
    appvols=hr_dv03,hr_dv04
```

- Create the Primary SRL.

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

- Create the Primary RVG.

```
# vradmin -g hrdg createpri hr_rvg \
    hr_dv01,hr_dv02,hr_vset01 hr_srl
```

- Uncomment the plus symbol from `/etc/vx/vras/.rdg` file from the nodes in the secondary site.
- 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.
- 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. IPv6 addresses are supported.

- Start replication.

```
# vradmim -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 and the volume set hr_vset01 on the seattle site.

```
# mkdir /hr_mount01
# mkdir /hr_mount02
# mkdir /hr_mount03
```

- 4 On seattle1, create file systems on the volumes hr_dv01 and hr_dv02 and on the volume set hr_vset01.

Verifying the VVR replication state

Test the replication state between seattle1 and london1 to verify that VVR is configured correctly.

To verify the replication state

- 1 Type the following command on each node:

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

- 2 In the output, verify the following:

- State of the RVG is `ENABLED/ACTIVE`.
- State of the RLINK is `CONNECT/ACTIVE`.

Configuring the agents for failover applications

This section explains how to configure the VVR agents for failover applications. See [“Configuring the agents for parallel applications”](#) on page 52.

Configure the RVG agent and RVGPrimary agent either when VCS is stopped or 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.

Use one of the following procedures to add the RVG resource to the VCS configuration:

- See “[Configuring the agents when VCS is running](#)” on page 45.
- See “[Configuring the agents when VCS is stopped](#)” on page 51.

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.

See “[Example configuration for a failover application](#)” on page 37.

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

To add the agent resources to your existing VCS configuration when VCS is running, perform the following procedures:

- Create the replication service group
- Create the application service group
- Create a dependency between the application service group and the replication service group
- Add an application service group and a replication service group to a composite service group for global failover

Perform the following steps on the client system `seattle1` in the Primary cluster Seattle, and then repeat the steps (with minor changes as noted) on the client system `london1` in Secondary cluster London:

To create the replication service group

- 1 Log in as root.
- 2 Set the VCS configuration mode to read/write by issuing the following command:

```
# haconf -makerw
```

- 3** Add the replication service group, `VVRGrp`, to the cluster. This group will contain all the storage and replication resources. Modify the attributes `SystemList` and `AutoStartList` of the service group to populate `SystemList` and `AutoStartList`:

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

On the Secondary cluster, replace `seattle1` and `seattle2` with `london1` and `london2`

- 4** 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
```

- 5** 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 lan3
```

- 6** 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 lan3
# hares -modify vvrip Address 192.168.40.20
# hares -modify vvrip NetMask "255.255.248.0"
```

On the Secondary cluster, use the appropriate IP for the Address. For example:

```
# hares -modify vvrip Address 192.168.40.21
```

- 7** Add the `RVG` resource `Hr_Rvg` in the `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
```

- 8 Specify resource dependencies for the resources you added in the previous steps:

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

- 9 Enable all resources in VVRGrp

```
# hagr -enableresources VVRGrp
```

- 10 Save and close the VCS configuration

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

Perform the following steps on the system `seattle1` in the Primary cluster `Seattle`, and then repeat the steps (with minor changes as noted) on the system `london1` in Secondary cluster `London`:

To create the application service group

- 1 Log in as root.
- 2 Set the VCS configuration mode to read/write by issuing the following command:

```
# haconf -makerw
```

- 3 Add a service group, `ORAGrp`, to the cluster `Seattle`. This group will contain all the application specific resources. Populate the attributes `SystemList`, `AutoStartList` and `ClusterList` of the service group.

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

On the Secondary, replace `seattle1` and `seattle2` with `london1` and `london2`, as follows:

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

- 4 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 lan0
```

- 5 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 lan0
# hares -modify oraip Address 192.168.40.1
# hares -modify oraip NetMask "255.255.248.0"
```

On the Secondary, modify the Address attribute for the IP resource appropriately.

- 6 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
```

- 7 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
```


8 Add the Mount resource Hr_Mount03 to mount the volume set hr_vset01 in the RVG resource Hr_Rvg:

```
# hares -add Hr_Mount03 Mount ORAGrp
# hares -modify Hr_Mount03 MountPoint /hr_mount03
# hares -modify Hr_Mount03 BlockDevice /dev/vx/dsk/ Hr_Dg/hr_vset01
# hares -modify Hr_Mount03 FSType vxfs
# hares -modify Hr_Mount03 FsckOpt %-n
# hares -modify Hr_Mount03 MountOpt rw
```

9 Add the Oracle resource Hr_Oracle

```
# hares -add Hr_Oracle Oracle ORAGrp
# hares -modify Hr_Oracle Sid hrl
# hares -modify Hr_Oracle Owner oracle
# hares -modify Hr_Oracle Home "/hr_mount01/OraHome1"
# hares -modify Hr_Oracle Pfile "inithrl.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
```

10 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"
```

11 Add the RVGPrimary resource Hr_RvgPri

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

- 12** 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 Hr_Mount03 rvg-pri
# hares -link oraip oranic
```

- 13** The application service group and the replication service group must both exist before doing this step. If you have not yet created the replication service group, do so now.

See [“Configuring the agents when VCS is running”](#) on page 45.

After you have created the application service group and the replication service group, specify an online local hard group dependency between `ORAGrp` and `VVRGrp`.

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

- 14** Enable all resources in `ORAGrp`

```
# hagrps -enableresources ORAGrp
```

- 15** Save and close the VCS configuration

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

- 16** Bring the service groups online, if not already online.

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

- 17** Verify that the service group `ORAGrp` is `ONLINE` on the system `seattle1` by issuing the following command:

```
# hagrps -state ORAGrp
```

Refer to the `/var/VRTSvcs/engine.log` file to identify configuration errors and to take actions accordingly.

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.

To configure the agents when VCS is stopped

- 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:

```
# hstop -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`:

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

- 8 Start VCS on all systems in both clusters.
- 9 Administer the service groups.

See [“Administering the service groups”](#) on page 59.

Configuring the agents for parallel applications

Use the RVGShared, RVGSharedPri, and the RVGLogowner agents to manage and monitor RVGs used by parallel applications in a shared environment.

Note: Determine the node that is performing the most writes by running the vxstat command on each node for a suitable period of time; after you set up replication, specify this node as the logowner.

The prerequisites for configuring the agents are as follows:

- You must have replication set up between the Primary and Secondary sites. For more information about replicating in a shared environment, see the *Veritas Volume Replicator Administrator's Guide*.
- The sites must be configured in a global cluster and the application service must be configured as a global service group. For more information about configuring global clusters, see the *Veritas Cluster Server Administrator's Guide*.

Sample configuration files are located in the /etc/VRTSvc/conf/sample_rac/ directory and include CVR in the filename. These sample files are installed as part of the VRTSdbac package, and can be used as a guide when creating your configuration. You can configure agents from the command line or from the VCS Java and Web consoles.

See the *Veritas Cluster Server Administrator's Guide* for more information.

To modify the VCS configuration on the Primary cluster

- 1 Define two new service groups: A logowner group that includes the RVGLogowner resource, and an RVG group that includes the RVGShared resource replication objects.
- 2 In the logowner group, define IP and NIC resources, used by the RLINKs for the RVG, and the RVGLogowner resource, for which the RVG and its associated disk group are defined as attributes.
- 3 In the RVG service group, set up the RVGShared agent to monitor the RVG resource. Because it is shared, the RVG must be configured to depend on the CVMVolDg resource, which defines the shared disk group and its activation mode.

Define the RVGShared and CVMVolDg resources within a parallel service group so that the service group may be online at the same time on all cluster nodes.

- 4 Add the RVGSharedPri resource to the existing application service group and define the service group to be a global group.
See the *Veritas Cluster Server Administrator's Guide* for instructions on how to create global groups.
- 5 Move the CVMVolDg resource from the existing application service group to the newly created RVGShared service group.
- 6 Set the following service group dependencies:
 - The RVG logowner service group has an “online local firm” dependency on the service group containing the RVG.
 - The RVG service group has an “online local firm” dependency on the CVM service group.
 - The application service group has an “online local firm” dependency on the RVG service group.

To modify the VCS configuration on the Secondary cluster

- 1 Log on to a node in the secondary cluster 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 Ensure VCS is not running while you edit main.cf by using the `hastop` command to stop the VCS engine on all systems and leave the resources available:

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

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

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

- 5 Use `vi` or another text editor to edit the main.cf file, making the following changes:
 - Edit the CVM group on the secondary cluster. Use the CVM group on the primary as your guide.

- Add the logowner group and the RVG service groups.
- Add an application service group. Use the application service group on the primary cluster as a pattern for the service group on the secondary cluster.
- Since the service group is a global group, assign it the same name as the group on the primary cluster.
- Define the ClusterList and ClusterFailOverPolicy cluster attributes.
- Include the RVGSharedPri resource.

6 Save and close the main.cf file.

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

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

8 Start VCS on all systems in both clusters.

The application group should be online on both systems of the primary cluster.

The application service group should not be online on the secondary cluster, but the CVM, RVG logowner, and RVG groups should be online.

Configuring the agents for a bunker replication configuration

This section describes how to set up the VCS agents for a bunker replication configuration, that is, an RDS that includes a bunker site. A bunker can be set up using the STORAGE protocol, or using IP.

Refer to one of the following sections to configure the VCS agents:

- See [“VCS configuration for a bunker using the STORAGE protocol”](#) on page 54.
- See [“VCS configuration for a bunker using IP”](#) on page 57.

VCS configuration for a bunker using the STORAGE protocol

When a bunker is set up using the STORAGE protocol, the disk group containing the bunker RVG is imported on the Primary node. If the Primary RVG is in a VCS cluster, the bunker RVG must remain online on the same node on which the Primary RVG is online.

In a shared disk group environment, the bunker RVG must be online on the logowner node.

This section describes how to configure the agents to automate the failover of the bunker RVG.

In a private disk group environment, the RVG resource handles the failover process. If the node on which the RVG resource is online fails, the RVG resource fails over to another node within the cluster. The RVG resource ensures that the bunker RVG also fails over, so that the bunker RVG continues to be on the same node with the Primary RVG.

In a shared disk group environment, the RVGLogowner agent handles the failover of the bunker RVG. If the logowner fails over, the bunker RVG must be deported from the original logowner node and imported on the new logowner node.

To set up automated failover of the bunker RVG, specify the bunker RVG, the bunker disk group, and the bunker node using the following attributes of the RVG resource in the application service group or the RVGLogowner agent:

Table 2-1 Attributes for configuring bunker failover

Attribute	Description
StorageDG	The name of the bunker disk group.
StorageRVG	The name of the bunker RVG.
StorageHostIds	Hostid of the bunker node or, if the bunker is clustered, a space-separated list of the hostids of each node in the bunker cluster.

The bunker failover attributes described in this section are the only specific attributes that differ for an RDS containing a bunker. The rest of the configuration for the VCSAgent is the same as for any other RDS.

See [“Example—setting up VVR in a VCS environment”](#) on page 40.

Sample configuration files for VCS agents in a bunker replication environment

The following examples show sample configuration files when the bunker Secondary is connected to the Primary using the STORAGE protocol.

This example uses the following names:

- seattle: primary cluster node
- london: bunker node
- bdg : bunker disk group name
- brvg: bunker RVG name

Sample configuration file (failover application)

The following sample file shows the configuration for the VCS agent on the Primary. The RVG agent includes attributes for a STORAGE bunker, to enable the bunker diskgroup to failover together with the parent RVG.

In this example, the disk group on the Primary is not a shared disk group.

If the Secondary for the RDS has a bunker associated to it, the RVG agent on the Secondary similarly would include the StorageRVG, StorageDG, and StorageHostIds attributes.

```
group AppSG (
    ClusterList = { cluster_london = 0 }
    SystemList = { seattle = 0, london = 1 }
    Authority = 1
    AutoStartList = { seattle }
    ClusterFailOverPolicy = Manual
)
    RVG RVG-1 (
RVG = vcsvrg
DiskGroup = pdg
Primary = true
StorageRVG = brvg
StorageDG = bdg
StorageHostIds = "portland"
    )
...

```

Sample configuration file (parallel application)

The following sample file shows the configuration for the VCS agent on the Primary. The RVGLogowner agent includes attributes for a STORAGE bunker, to enable the bunker diskgroup to failover together with the logowner. In this example, the disk group on the Primary is a shared disk group. If the Secondary for the RDS has a bunker associated to it, the RVGLogowner resource on the Secondary similarly would include the StorageRVG, StorageDG, and StorageHostIds attributes.

```
group RVGLogownerGrp (
    SystemList = { seattle = 0, london = 1 }
    AutoStartList = { seattle, london }
)
    IP vvr_ip (
        Device = bge0
        Address = "192.168.3.13"
    )

```



```
    )  
    NIC vvr_nic (  
        Device = bge0  
    )  
    RVGLogowner vvr_rvglogowner (  
    RVG = rvg  
    DiskGroup = vvr dg  
    StorageRVG = brvg  
    StorageDG = bdg  
    StorageHostIds = "portland"  
    )  
requires group RVGSharedGrp online local firm  
    vvr_ip requires vvr_nic
```

VCS configuration for a bunker using IP

The configuration for the VCS agents for a bunker over IP is the same as for any other Secondary.

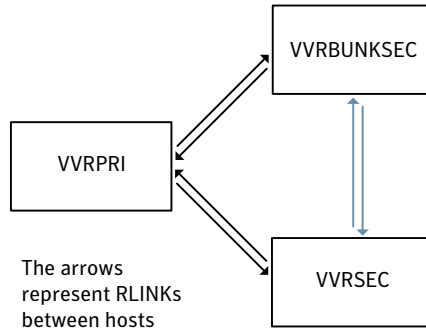
To set up a bunker configuration

- 1 The Primary and Secondary configurations are the same as for any other RDS using VCS agents.
See [“Example—setting up VVR in a VCS environment”](#) on page 40.
- 2 Add the bunker to the RDS with the `vradmin addbunker` command.
For a detailed procedure, see the *Veritas Volume Replicator Administrator's Guide*.
- 3 Configure the VCS agents on the secondary in the same way as the configuration for any other Secondary. There is no special configuration that needs to be done for a bunker over IP.

How the RVGPrimary works in a bunker setup

Under normal operating conditions, the Primary cluster (VVRPRI) site replicates data to the bunker's Secondary (VVRSEC) site in the synchronous override mode. It replicates the data in the synchronous mode to ensure that it is up-to-date.

Figure 2-4 RLINKs among Primary, Secondary, and bunker hosts, where the arrows represent RLINKs between hosts



If a disaster occurs at the VVRPRI, the RVGPrimary agent on VVRSEC activates the Bunker node and starts replay from the Bunker Replicator Log to VVRSEC. During this replay, the agent converts the Bunker node to a Primary and the agent uses the data in the Replicator Log to bring the Secondary up-to-date.

When the replay completes or the BunkerSyncTimeout attribute's timeout limit has elapsed, the Secondary takes over the Primary's role and deactivates the Bunker node.

For a Storage Bunker configuration, if a disaster occurs at the Primary then the RVGPrimary agent comes online on the secondary node VVRSEC. The agent first imports the disk group on the Bunker node and then it activates the Bunker node to start replay to the Secondary.

When the original Primary becomes available, you can migrate the Primary role back to the original site. If you had performed the takeover with auto failback then you have enabled failback logging at takeover.

If the original Primary becomes available again, the agent converts it to the Secondary. The new Primary writes to the original Primary to bring it up-to-date. After a successful migration or takeover, the agent tries to start the replication to the Bunker if any bunker sites are associated with the new Primary. If this operation fails, an VCS logs an error message in its engine log. After you review the error message, you can then start the replication to bunker outside of VCS.

About bunker SRL overflow

The bunker replay does not occur if the bunker SRL has overflowed or the bunker SRL is not in useful state. In the case of an overflowed bunker or one that is in a useless state, the RVGPrimary agent logs an error message while attempting to initiate the bunker replay and then faults the resource. Since the bunker replay cannot happen in this situation, Symantec recommends that you set the

BunkerSyncTimeout value to 0. A value of 0 means that you do not choose to use the bunker replay and that the online operation can be completed successfully without bunker replay.

About global clustering with bunker replay

When the value of the ClusterFailoverPolicy attribute is Auto and the Application service group is configured on some nodes of the primary cluster, global clustering immediately detects any system fault at the primary site and fails over the Application service group to the remote site. VVR might take longer to detect the fault at the primary site and to complete its configuration changes to reflect the fault. To ensure that global clustering successfully initiates a bunker replay, Symantec recommends that you set the value of the OnlineRetryLimit attribute to a non-zero value for the RVGPrimary resource when the primary site has a bunker configured.

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.

To administer a VCS service group

- 1 Start the VCS engine on seattle1:

```
# hastart
```

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

```
# hagr -display
```

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

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

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

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

- 5** Start the VCS engine on seattle2:

```
# hstart
```

- 6** Switch the VVR service group to seattle2:

```
# hagrps -switch hr_grp -to seattle2
```

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

```
# hagrps -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
```

Sample main.cf files

This appendix includes the following topics:

- [Globally clustered VCS and VVR main.cf](#)

Globally clustered VCS and VVR main.cf

The following main.cf file applies to a globally clustered VCS and VVR configuration. It is common for the the elect-primary feature and for use with bunker with the global cluster options with VVR.

```
include "OracleASMTypes.cf"
include "types.cf"
include "Db2udbTypes.cf"
include "OracleTypes.cf"
include "SybaseTypes.cf"

cluster gco2 (
    UserNames = { admin = dqrJqlQnrMrrPzrLqo }
    ClusterAddress = "10.182.44.221"
    Administrators = { admin }
)

remoteclass gco1 (
    ClusterAddress = "10.182.71.20"
)

heartbeat Icmp (
    ClusterList = { gco1 }
    StopTimeout @gco1 = 60
    AYATimeout @gco1 = 300
    AYARetryLimit = 1
)
```

```
        Arguments @gcol = { "10.182.71.20" }
    )

system msdn15 (
)

system msdn16 (
)

group ClusterService (
    SystemList = { msdn15 = 0, msdn16 = 1 }
    AutoStartList = { msdn15, msdn16 }
    OnlineRetryLimit = 3
    OnlineRetryInterval = 120
)

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

IP webip (
    Device = nge0
    Address = "10.182.44.221"
    NetMask = "255.255.240.0"
)

NIC csgnic (
    Device = nge0
)

wac requires webip
webip requires csgnic

// resource dependency tree
//
//     group ClusterService
//     {
//     Application wac
//     {
```

```
//          IP webip
//          {
//          NIC csgrnic
//          }
//          }
//          }

group VVRGrp (
    SystemList = { msdn15 = 0, msdn16 = 1 }
    AutoStartList = { msdn15, msdn16 }
)

DiskGroup Hr_Dg (
    DiskGroup = dg1
)

IP vvrrip (
    Device = nge0
    Address = "10.182.44.220"
    NetMask = "255.255.240.0"
)

NIC vvrnic (
    Device = nge0
)

RVG Hr_Rvg (
    RVG = rvg1
    DiskGroup = dg1
)

Hr_Rvg requires Hr_Dg
Hr_Rvg requires vvrrip
vvrrip requires vvrnic

// resource dependency tree
//
//          group VVRGrp
//          {
//          RVG Hr_Rvg
//          {
```

```
//          DiskGroup Hr_Dg
//          IP vvrip
//          {
//          NIC vvrnic
//          }
//          }
//          }

group app_rep (
    SystemList = { msdn16 = 0, msdn15 = 1 }
    ClusterList = { gco1 = 1, gco2 = 0 }
    AutoStartList = { msdn16, msdn15 }
    ClusterFailOverPolicy = Auto
)

FileOnOff test-res (
    PathName = "/var/tmp/test"
)

RVGPrimary rvg_prim_res (
    RvgResourceName = Hr_Rvg
    AutoResync = 1
)

requires group VVRGrp online local hard
test-res requires rvg_prim_res

// resource dependency tree
//
//          group app_rep
//          {
//          FileOnOff test-res
//          {
//          RVGPrimary rvg_prim_res
//          }
//          }
//          }

group app_rep_fd (
    SystemList = { msdn16 = 0, msdn15 = 1 }
    UserStrGlobal = "FD:app_rep"
```



```
AutoFailOver = 0
TypeDependencies = { IP, Mount, RVGSnapshot }
)

IP webip_fd (
    Device = nge0
    Address = "10.182.44.221"
    NetMask = "255.255.240.0"
)

RVGSnapshot Hr_Rvg-sos (
    RvgResourceName = Hr_Rvg
    CacheObj = cacheobj
    Prefix = snap
)

requires group VVRGrp online local firm

// resource dependency tree
//
//     group app_rep_fd
//     {
//     RVGSnapshot Hr_Rvg-sos
//     IP webip_fd
//     }
```


Index

A

- agents. *See* individual agents. *See* RVG agent
 - best practices for setting up 36
 - configuring 44
 - configuring when VCS is stopped 51
 - list of VVR 9
 - RVGLogowner. *See* RVGLogowner agent
 - RVGPrimary. *See* RVGPrimary agent
 - RVGShared. *See* RVGShared agent
 - RVGSharedPri. *See* RVGSharedPri agent
 - RVGSnapshot. *See* RVGSnapshot agent
 - setting up
 - best practices 36
- AutoResync attribute
 - RVGPrimary agent 30

B

- best practices
 - setting up VVR agents 36

C

- configuration
 - setting up the VVR 41
- configuring RVG agent
 - when VCS is started 45
 - when VCS is stopped 51
- configuring VVR in VCS environment
 - overview 32
 - requirements 35

D

- dependency graphs
 - RVGLogowner agent 25
 - RVGPrimary agent 14
 - RVGShared agent 23
 - RVGSharedPri agent 29

E

- examples
 - setting up VVR in a VCS environment 40

F

- failover group
 - RVGLogowner agent 25
- fast failback
 - AutoResync attribute of RVGPrimary 30
- fast failback resynchronization
 - RVGPrimary 13
 - RVGSharedPri 29
- fire drill
 - RVGSnapshot agent 19

G

- generic VVR setup in a VCS environment 33

H

- hybrid group
 - about 31

L

- list of agents for VVR 9
- logowner
 - virtual IP requirement 25

M

- migrating
 - RVGPrimary 13
 - RVGSharedPri 28
- Mount resource
 - volume sets 36

N

- noautoimport attribute
 - RVG agent requirement 36

O

- overview
 - configuring VVR in a VCS environment 32

- P**
- parallel group
 - RVGShared agent 23
- R**
- RDC
 - about 31
 - SystemZones attribute of RVG and RVG Primary agents 31
 - Replicated Data Cluster. *See* RDC
 - replication
 - setting up 35
 - replication state
 - verifying 44
 - requirements
 - configuring VVR in VCS environment 35
 - RVG agent
 - configuring when VCS is started 45
 - configuring when VCS is stopped 51
 - described 10
 - requirement
 - noautoimport 36
 - SystemZones attribute 31
 - virtual IP requirement 43
 - RVGLogowner agent
 - configuring 52
 - dependency graph 25
 - described 25
 - failover group 25
 - RVGPrimary agent
 - dependency graph 14
 - described 13
 - migrating 13
 - SystemZones attribute 31
 - takeover 13
 - RVGShared agent
 - configuring 52
 - dependency graph 23
 - described 22
 - parallel group 23
 - RVGSharedPri agent
 - configuring 52
 - dependency graph 29
 - described 28
 - migrating 28
 - takeover 28
 - RVGSnapshot agent
 - described 19
 - fire drill 19
- S**
- sample configuration files
 - to configure agent
 - location 41
 - setting
 - noautoimport attribute 36
 - setting up
 - replication 35
 - setting up the VVR configuration 41
 - setting up VVR agents
 - best practices 36
 - snapshots
 - using RVGSnapshot agent for 19
 - state of replication 44
 - SystemZones attribute of RVG and RVG Primary agents 31
- T**
- takeover
 - RVGPrimary 13
 - RVGSharedPri 28
- V**
- VCS
 - configuring RVG agent with 45, 51
 - VCS agents for VVR
 - list 9
 - VCS environment
 - configuring VVR in 32
 - example setting up VVR 40
 - generic VVR setup 33
 - requirements for configuring VVR 35
 - setting up VVR
 - virtual IP requirement 43
 - verifying
 - VVR replication state 44
 - virtual IP
 - requirement 43
 - RVGLogowner agent requirement 25
 - volume sets
 - using agents with 36
 - VVR agents
 - configuring 44
 - list of 9
 - VVR configuration
 - setting up 41
 - VVR in a VCS environment
 - configuring 32

VVR in a VCS environment (*continued*)
 requirements 35
 set up example 40
 virtual IP requirement for setting up 43
VVR setup in a VCS environment 33