VERITAS™

VERITAS Volume Replicator 4.1 Support for HP Serviceguard

Administrator's Guide

HP-UX

N13529G

June 2005

Disclaimer

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

VERITAS Legal Notice

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

Third-Party Legal Notices

Data Encryption Standard (DES) Copyright

Copyright © 1990 Dennis Ferguson. All rights reserved.

Commercial use is permitted only if products that are derived from or include this software are made available for purchase and/or use in Canada. Otherwise, redistribution and use in source and binary forms are permitted.

Copyright 1985, 1986, 1987, 1988, 1990 by the Massachusetts Institute of Technology. All rights reserved.

Export of this software from the United States of America may require a specific license from the United States Government. It is the responsibility of any person or organization contemplating export to obtain such a license before exporting.

WITHIN THAT CONSTRAINT, permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of M.I.T. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. M.I.T. makes no representations about the suitability of this software for any purpose. It is provided as is without express or implied warranty.

Apache Software

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work.

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

Contents

Preface

The VERITAS Volume Replicator (VVR) Support for HP Serviceguard document guides you through the process of setting up VVR in an HP Serviceguard environment.

Audience

This guide is intended for system administrators who are responsible for installing, configuring, and setting up replication using VVR in an HP Serviceguard Environment. This guide assumes that the user has:

- A basic understanding of system administration.
- Good understanding of the VVR product.
- A good understanding of the clustering concepts.
- Fairly good understanding of the HP Serviceguard product.

How This Guide Is Organized

Chapter 1, "Introduction" on page 1 provides an overview of VVR support for HP Serviceguard.

Chapter 2, "Installing the VRTSvrmcsg Package" on page 3 explains the procedure to install the package to enable VVR support for HP Serviceguard.

Chapter 3, "Understanding the Functions in the vvr.sh Script" on page 9 provides an overview of the `vvr.sh` script that enables you to monitor the RVG.

Chapter 4, "Configuring VVR in an HP Serviceguard Environment" on page 13 describes the procedure to create and modify the control and configuration scripts.

Chapter 5, "Testing the Package Switching and Failover" on page 33 describes the procedure to test the package switching and failover.

Related VERITAS Documents

For more information on any of the topics presented in this guide, refer to the VERITAS Volume Replicator (VVR) or the VERITAS Volume Manager (VxVM) documentation sets. Refer to the *VERITAS Volume Replicator Release Notes* for more information on these 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 x.
#	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.

Introduction 1

HP Serviceguard is a specialized facility for protecting mission-critical applications from a wide variety of hardware and software failures. With HP Serviceguard, multiple nodes (systems) are organized into an enterprise cluster that delivers highly available application services to LAN-attached clients. HP Serviceguard monitors the health of each node and quickly responds to failures in a way that minimizes or eliminates application downtime.

VERITAS Volume Replicator (VVR) support for HP Serviceguard enables you to set up an efficient and reliable failover support for VVR on HP-UX. In addition to configuring the cluster, you also need to identify the applications and services that you want to group into packages. Packages are the means by which Serviceguard starts and halts configured applications. A package is a collection of services, disk volumes and IP addresses that are managed by Serviceguard to make sure they are available.

The package control scripts contain all the information necessary to run all the services in the package, monitor them during operation, respond to a failure, and halt the package when necessary. These scripts are modified to include information about VVR. The vvr.sh script provides the functionality to manage and monitor VVR RVG resource. The package control scripts use this functionality provided by the `vvr.sh` script to manage the VVR resources along with volume groups and disk groups. For more information on the functions that the `vvr.sh` script contains, see Chapter 3, "Understanding the Functions in the vvr.sh Script" on page 9.

This document discusses some of the basics on installing, configuring, and testing failover of VVR within an HP Serviceguard cluster.

Note To achieve high availability across both sites, contact your VERITAS Sales Representative for additional information on VERITAS Disaster Recovery and High Availability products.

Installing the VRTSvrmcsg Package 2

This chapter discusses the procedure to install the VVR support for HP Serviceguard package.

Prerequisites

- ✔ Install VxVM, which also installs VVR. If necessary, refer to the *VERITAS Volume Replicator Installation Guide*.
- ✔ Obtain licenses for the VERITAS Volume Manager (VxVM) and VVR.
- ✔ VVR support for HP Serviceguard package does not require a separate license. However, the licenses for both VVR and HP Serviceguard must be present.

Additional Requirements

- ✔ Ensure that the correct version of HP-UX, VVR and HP Serviceguard are installed on the appropriate nodes. For more details on the versions refer to the *VERITAS Volume Replicator Release Notes*.

Installing VVR

VVR is installed as a part of the VxVM installation process. For further details on installing VVR refer to the *VERITAS Volume Replicator Installation Guide*.

Note The VVR executables should not be shared, but should be accessible from the same local path on each node.

Installing the VRTSvrmcsg Package

This section describes the procedure to install `VRTSvrmcsg` package, using either the command line or the GUI.

Before Installing the VRTSvrmcsg Package

To install the package through a graphical user interface on each system, perform the following steps.

1. Log in as `root`.

2. Set the display permission on your workstation:

   ```
   # xhost +<myws>
   ```

3. Configure the shell environment variable `DISPLAY` to your workstation. For example, if your workstation has the name **myws**, type:

   For Bourne or Korn shell (sh or ksh):

   ```
   # DISPLAY=myws:0.0
   # export DISPLAY
   ```

   For C shell (csh or tcsh):

   ```
   # setenv DISPLAY myws:0.0
   ```

   Setting the DISPLAY environment variable in this way enables you to use the SD-UX graphical interface.

Proceed to the following section if each system in the cluster has a DVD-ROM drive. Otherwise, proceed to the section “Installing the VRTSvrmcsg Package From a Software Depot” on page 6.

Mounting the DVD-ROM

1. Insert the disc into a drive connected to your system.

2. Log in as `root`.

3. Create a mount point directory, `/dvdrom`, if it does not exist. The directory must have read/write permissions.

4. Determine the block device file for the DVD-ROM drive:

   ```
   # ioscan -fnC disk
   ```

   For example, the listing may indicate the DVD-ROM drive's block device is `/dev/dsk/c1t2d0`; make a note of the device file as it applies to your system.

   Note The format for the device access name is `cxtydz`, where `x` is the controller number, `y` is the target number, and *`z`* is the device number.

5. Mount the software disc. For example, to mount the disc having block device `/dev/dsk/c1t2d0` to the mount point `/dvdrom`, enter:

   ```
   # mount -o ro /dev/dsk/c1t2d0 /dvdrom
   ```

   To list the packages on the DVD-ROM:

   ```
   # cd /dvdrom/depot
   # ls
   ```

Installing the VRTSvrmcsg Package From the Software Disc

If each system in the cluster has a DVD-ROM drive, use the following steps to install the package on each system in the cluster.

You must perform these steps on every node that is intended to be a part of the cluster.

1. If you have not mounted the DVD-ROM drive follow the procedure given in the section "Mounting the DVD-ROM" on page 5.

2. Use the swinstall command to install the package.

   ```
   # swinstall -s /dvdrom/depot VRTSvrmcsg
   ```

   Note The **`VRTSvrmcsg`** package will be installed in the location `/opt/VRTSvrmcsg`. The `VRTSvrmcsg` directory is created during installation.

3. When the installation is complete, verify that the package is installed, by using the following command:

   ```
   # swlist | grep VRTSvrmcsg
   ```

4. Repeat step 1 to step 3 on each node that is intended to be a part of the cluster.

Installing the VRTSvrmcsg Package From a Software Depot

This method of installing the package is used when only one node in the cluster has a DVD-ROM drive or when you are installing to a cluster with several machines. In this case use the SD-UX swcopy command to copy the package from the software disc to a software depot on the same node. From that depot, you can install the package on each node in the cluster over the local network.

1. Install **VRTSvrmcsg** package on the node in the cluster with a DVD-ROM using the procedure given in the section, "Installing the VRTSvrmcsg Package From the Software Disc" on page 5.

2. On the same node on which the software disc is mounted, start the SD-UX user interface to copy the package to a depot.

   To copy the packages to the /var/spool/sw directory, use the following command:

   ```
   # swcopy -s /dvdrom/depot
   ```

   If the depot name is different then use the following command:

   ```
   # swcopy -s /var/spool/<depot_name>
   ```

3. Click **OK**.

4. To copy the package from the Software Selection screen, perform the following steps:

   a. Highlight **VRTSvrmcsg** and choose the Mark for Copy option from the Actions menu.

   b. After marking the package for copying, choose the Copy option from the Actions menu.

   c. When the copy analysis is complete, click **OK**.

   d. Choose **Done** on the confirmation screen when you see that the copying is complete.

   e. Exit SD-UX.

5. On each of the other nodes login as `root`.

6. Install the package from the depot by typing:

   ```
   # swinstall -s <depothost>:/var/spool/sw VRTSvrmcsg
   ```

7. Verify that the package is installed correctly by using the following command:

   ```
   # swlist | grep VRTSvrmcsg
   ```

8. Repeat step 5 to step 7 to install the package on each of the nodes.

Removing the Package

Type the following command on each node to remove the package.

```
# swremove VRTSvrmcsg
```

Understanding the Functions in the vvr.sh Script 3

This chapter describes how you can use the `vvr.sh` script to start, monitor, and stop the VVR RVG resource.

The package control scripts contain all the information necessary to run all the services in the package, monitor them during operation, respond to a failure, and halt the package when necessary. These scripts are modified to include information about VVR.

The vvr.sh script provides the functionality to manage and monitor VVR RVG resource. This script provides specific functions that enable you to perform the required tasks. The package control scripts use this functionality provided by the `vvr.sh` script to manage the VVR resources along with volume groups and disk groups.

Starting the RVG

The `vvr.sh` script provides a function `vvr_activate_cmds` that performs the following tasks for the specified disk groups:

- Starts all the VVR volumes under the RVG.
- Starts the SRL volume for the specified RVG.
- Recovers the RVGs and RLINKs within the specified disk groups.
- Starts the RVGs.

Usage

```
/opt/VRTSvrmcsg/vvr.sh [-r max_retry] activate DiskGroup(s)
```

Option	Description
`-r` *`max_retry`*	Specifies the maximum number of times the script must retry starting the volumes and SRL under an RVG in case of a failure. The default is 5.

Example

```
/opt/VRTSvrmcsg/vvr.sh activate hrdg
/opt/VRTSvrmcsg/vvr.sh -r 2 activate hrdg

/opt/VRTSvrmcsg/vvr.sh activate hrdg1 hrdg2
/opt/VRTSvrmcsg/vvr.sh -r 3 activate hrdg1 hrdg2
```

Monitoring the RVG

The `vvr.sh` script provides a function `vvr_monitor_cmds` that monitors the state of all the RVGs in the specified disk groups. It checks for the following RVG states:

- `ENABLED / EMPTY`
- `ENABLED / ACTIVE`

It also monitors the `vradmind` and `in.vxrsyncd` daemons.

Usage

`/opt/VRTSvrmcsg/vvr.sh [-i` *`interval`*`] monitor DiskGroup(s)`

Option	Description
`-i` *`interval`*	Specifies the time interval in seconds, during which the script will wait between checks, for the RVG state. The default is 30 seconds.

An RVG is monitored for local access only. Replication is not monitored. The replication fails if the RVG is not in the ENABLED state.

Examples

```
/opt/VRTSvrmcsg/vvr.sh monitor hrdg
/opt/VRTSvrmcsg/vvr.sh -i 10 monitor hrdg

/opt/VRTSvrmcsg/vvr.sh monitor hrdg1 hrdg2
/opt/VRTSvrmcsg/vvr.sh -i 50 monitor hrdg1 hrdg2
```

Stopping the RVG

The vvr.sh script provides a function vvr_halt_cmds that stops all the RVGs in the specified disk group(s).

Usage

```
/opt/VRTSvrmcsg/vvr.sh halt DiskGroup(s)
```

Examples

```
/opt/VRTSvrmcsg/vvr.sh halt hrdg
/opt/VRTSvrmcsg/vvr.sh halt hrdg1 hrdg2
```

Configuring VVR in an HP Serviceguard Environment 4

This chapter describes the procedure to create and modify the control and configuration scripts appropriately so that the VVR support for HP Serviceguard can be used effectively. Before you start configuring the support package, refer to "Understanding the Functions in the vvr.sh Script" on page 9 for information on the `vvr.sh` script.

Note The entire process for configuring and setting up the support package has been explained with reference to a sample configuration. Replace the various parameter values shown with appropriate values to suit your environment.

Prerequisites

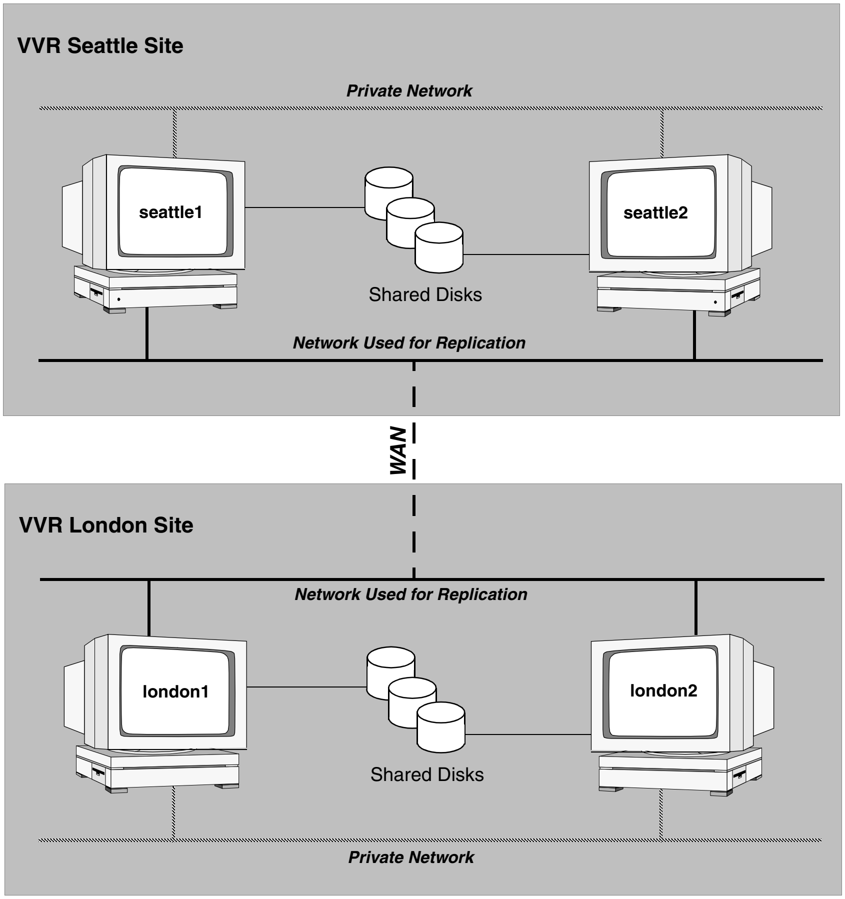
- ✔ Make sure that the volumes that are a part of the RVG are present on the shared storage.
- ✔ Make sure the disk group can be deported and imported on all the nodes in the cluster.
- ✔ VVR must use a virtual IP address for replication.
- ✔ All the nodes in the VVR cluster environment must use the same port number for replication. You can verify this by using the `vrport` command.

  For more information on displaying and changing the port numbers, see section "Displaying and Changing the Ports Used by VVR" in the *VERITAS Volume Replicator Administrator's Guide*.

Sample VVR Configuration

In the following example, two clusters are located at separate sites. Each site serves as a disaster recovery site for the other. VVR replicates data between the sites via a WAN.

The first cluster is located in Seattle, and consists of two nodes: `seattle1` and `seattle2`. The second cluster is located in London, and also consists of two nodes: `london1` and `london2`. The nodes located at the Seattle site contain the Primary RVG, the nodes located at the London site contain the Secondary RVG.



Setting Up the VVR Configuration

This section details the procedure to set up the VVR configuration using the Sample VVR Configuration. The following table outlines the names that are used by the sample configuration. For more information on configuring VVR, refer to the *VERITAS Volume Replicator Administrator's Guide*.

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	`192.2.40.74`

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	`192.2.40.53`

Note The 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.

Checking the Deport and Import Operations

Perform the following steps to confirm whether the disk group hrdg can be successfully deported and imported.

1. On host seattle1 run the following command to deport the disk group:

   ```
   # vxdg deport hrdg
   ```

2. On host seattle2 run the following commands to import and deport the disk group:

   ```
   # vxdg -t import hrdg
   # vxdg deport hrdg
   ```

3. On host seattle1 run the following command to import the disk group:

   ```
   # vxdg -t import hrdg
   ```

Repeat the step 1 to step 3 on hosts london1 and london2 to ensure that the deport and import commands work correctly.

Note Also ensure that the /etc/vx/vras/.rdg file on both the Secondary hosts london1 and london2 contains the disk group ID of the Primary's disk group hrdg.

Creating the Secondary Data Volumes

Perform the following steps on host london1:

1. Create the Secondary data volumes.

   ```
   # vxassist -g hrdg make hr_dv01 4G \
       layout=mirror logtype=dcm mirror=2

   # vxassist -g hrdg make hr_dv02 4G \
       layout=mirror logtype=dcm mirror=2
   ```

2. Create the Secondary SRL.

   ```
   # vxassist -g hrdg make hr_srl 4G mirror=2
   ```

Creating the Primary Data Volumes

Perform the following steps on host `seattle1`:

1. Create the Primary data volumes.

```
# vxassist -g hrdg make hr_dv01 4G \
     layout=mirror logtype=dcm mirror=2

# vxassist -g hrdg make hr_dv02 4G \
     layout=mirror logtype=dcm mirror=2
```

2. Create the Primary SRL .

```
# vxassist -g hrdg make hr_srl 4G mirror=2
```

Creating the Primary and Secondary RVGs

Perform the following steps on host `seattle1`:

1. Create the Primary RVG.

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

2. Create the Secondary RVG.

```
# vradmin -g hrdg addsec hr_rvg 192.2.40.74 192.2.40.53 \
        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 192.2.40.74 must be able to ping virtual IP address 192.2.40.53 and vice versa.

Mounting the Volumes in the Primary RVG

Perform the steps given below to mount the volumes under the Primary RVG.

1. Create the mount points for the volumes by creating the following directories on `seattle1` and `seattle2`. These directories will be used as mount points for volumes `hr_dv01` and `hr_dv02` on the VVR Seattle site.

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

2. Create the file system on the volumes under the Primary RVG by using the following command on `seattle1`:

   ```
   # mkfs -F vxfs /dev/vx/rdsk/hrdg/hr_dv01
   # mkfs -F vxfs /dev/vx/rdsk/hrdg/hr_dv02
   ```

3. Mount the volumes in the Primary RVG by using the following command on `seattle1`:

   ```
   # mount /dev/vx/dsk/hrdg/hr_dv01 /hr_mount01
   # mount /dev/vx/dsk/hrdg/hr_dv02 /hr_mount02
   ```

Starting Replication

To start replication run the following command on the host `seattle1`:

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

Verifying the State of Replication

Verify the replication state between `seattle1` and `london1` to ensure that VVR is configured correctly. Type the following command on each of the nodes `seattle1` and `london1`:

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

In the output of this command verify that:

✔ The RVG `hr_rvg` is in the ENABLED/ACTIVE state.

✔ The RLINKs `rlk_london_hr_rvg` and `rlk_seattle_hr_rvg` are in the CONNECT/ACTIVE state.

Configuring the Cluster

This section describes the procedure to set up a cluster.

Perform the following steps on the node seattle1

1. Create the ASCII cluster template file.

   ```
   # cmquerycl -v -C /etc/cmcluster/cluster.ascii -n seattle1 \
       -n seattle2
   ```

2. Modify the `cluster.ascii` file to include your cluster environment details.

3. Verify the cluster configuration.

   ```
   # cmcheckconf -v -C /etc/cmcluster/cluster.ascii \
        -n seattle1 -n seattle2
   ```

4. Create the cluster by applying the configuration file.

   ```
   # cmapplyconf -v -C /etc/cmcluster/cluster.ascii
   ```

   This command will create the `cmclconfig` binary file and distribute it to the nodes `seattle1` and `seattle2`.

5. Start the cluster.

   ```
   # cmruncl -v -n seattle1 -n seattle2
   ```

6. Check the cluster status.

   ```
   # cmviewcl -v
   ```

7. Halt and restart the cluster using the following commands, to test whether it works correctly.

   Halt the cluster.

   ```
   # cmhaltcl -f -v
   ```

   Restart the cluster.

   ```
   # cmruncl -v -n seattle1 -n seattle2
   ```

8. Repeat step 1 to step 7 on node london1, replacing the node names `seattle1` by `london1` and `seattle2` by `london2`.

Note For more details on the cluster configuration, refer to the HP Serviceguard documentation available on the HP-UX documentation site.

Creating and Modifying the HP Serviceguard Package Configuration File

The package configuration file needs to be created and modified for both the Primary cluster (`seattle`) and Secondary cluster (`london`).

Creating and Modifying the Configuration File for the Primary Cluster

Perform the following steps on the node `seattle1`:

1. Create the package directory.

   ```
   # mkdir /etc/cmcluster/pri_pkg
   ```

2. Create the package configuration file.

   ```
   # cmmakepkg -p /etc/cmcluster/pri_pkg/pri_pkg.conf
   ```

   You can also do this using SAM.

   The configuration file that is generated by the `cmmakepkg` command is created with the parameters set to default values. You will need to customize this file to suit your requirements.

3. Edit the package configuration file, `pri_pkg.conf` using `vi` or any other editor.

   Refer to the following table for the parameters that you need to change in the package configuration file to configure VVR in an HP Serviceguard environment.

   A sample package configuration file, `pri_pkg.conf` is available in the `/opt/VRTSvrmcsg/samples/seattle` directory.

Parameter	Value
Specifying a Package Name Modify the `PACKAGE_NAME` parameter to specify the appropriate package name as indicated by the description in the file. This name should be a unique within a cluster.	
`PACKAGE_NAME`	`pri_pkg`
Specifying the Node Names You must specify the names of all the nodes in the cluster. It is not important to maintain the order.	

Parameter	Value
NODE_NAME	seattle1
NODE_NAME	seattle2

Specifying the path to the Run and Halt scripts

In most cases the run script and halt script specified here will be the same script, the package control script generated by the cmmakepkg command. This control script handles the running and halting of the package. Enter the timeout, specified in seconds, for the run and halt scripts. If the script has not completed by the specified timeout value, it will be terminated. The default for each script timeout is NO_TIMEOUT.

RUN_SCRIPT	/etc/cmcluster/pri_pkg/pri_pkg.cntl
RUN_SCRIPT_TIMEOUT	NO_TIMEOUT
HALT_SCRIPT	/etc/cmcluster/pri_pkg/pri_pkg.cntl
HALT_SCRIPT_TIMEOUT	NO_TIMEOUT

Specifying the Service Name

Specify the appropriate service related information by modifying the following parameters.

Note The service name must match the name used in the HP Serviceguard control script.

SERVICE_NAME	VVR
SERVICE_FAIL_FAST_ENABLED	NO
SERVICE_HALT_TIMEOUT	300

Specifying the Subnet Name

Enter the network subnet name that is to be monitored for this package. Repeat this line as required, for additional subnet names. If any of the subnets defined goes down, the package will be switched to another node that is configured for this package and has all the defined subnets available.

SUBNET	192.2.40.0

Save the changes you have made to the pri_pkg.conf file.

Creating and Modifying the Configuration File for the Secondary Cluster

Perform the following steps on the node london1:

1. Create the package directory.

   ```
   # mkdir /etc/cmcluster/sec_pkg
   ```

2. Create the package configuration file.

   ```
   # cmmakepkg -p /etc/cmcluster/sec_pkg/sec_pkg.conf
   ```

Modify the file as described in "Creating and Modifying the Configuration File for the Primary Cluster" on page 20. Replace the parameters appropriately with respect to the host `london1`. A sample package configuration file, `sec_pkg.conf` is available in the `/opt/VRTSvrmcsg/samples/london` directory.

Creating and Modifying the HP Serviceguard Package Control Script

The package control script needs to be created and modified for both the Primary (`Seattle`) and Secondary (`London`) cluster. The following sections describe the process separately for each cluster.

Creating and Modifying the Package Control Script for the Primary Cluster

Perform the following steps on the node `seattle1`:

1. Create the HP Serviceguard package control script by using the following command:

   ```
   # cmmakepkg -s /etc/cmcluster/pri_pkg/pri_pkg.cntl
   ```

   You can also do this using SAM.

   The control script that is generated by the `cmmakepkg` command is created with the parameters set to default values. Customize this script to suit your requirements. Edit the package control script as indicated by the comments in that file.

   Note For more information on HP Serviceguard refer to the appropriate documentation available on the HP-UX documentation site.

2. Edit the package control script using `vi` or any other editor.

   Refer to the following points for the parameters that you need to change in the control script to configure HP Serviceguard for your environment.

   A sample package control script `pri_pkg.cntl` is present in the `/opt/VRTSvrmcsg/samples/seattle` directory.

- Specifying Disk Group Information

  The `VXVM_DG` parameter is used to indicate the disk groups which are used by VVR.

  Example

  ```
  VXVM_DG[0]=”hrdg”
  ```

- Specifying File System Information

  File systems that are associated with the specified disk groups are defined as shown below.

Example

```
LV[0]="/dev/vx/dsk/hrdg/hr_dv01 "; FS[0]="/hr_mount01"; \
FS_MOUNT_OPT[0]="-o rw"; FS_UMOUNT_OPT[0]=""; \
FS_FSCK_OPT[0]=""; FS_TYPE[0]="vxfs"
LV[1]="/dev/vx/dsk/hrdg/hr_dv02"; FS[1]="/hr_mount02"; \
FS_MOUNT_OPT[1]="-o rw" ; FS_UMOUNT_OPT[1]=""; \
FS_FSCK_OPT[1]=""; FS_TYPE[1]="vxfs"
```

❖ Specifying the VVR Recovery Parameter

Use this parameter to specify recovery procedure if the RVG fails.

Specify the VVR recovery statement just before the volume recovery section in the script.

Example

```
VXRVG="/opt/VRTSvrmcsg/vvr.sh activate \$DiskGroup"
```

❖ Specifying the IP and Subnet

Use these parameters to specify the IP and subnet address of the host `seattle1`. Note that the IP that you specify must be the virtual IP used by the RLINK.

```
IP[0]="192.2.40.74"
SUBNET[0]="192.2.40.0"
```

❖ Specifying the Service Name Information

The service name must be the same as defined in the HP Serviceguard configuration file `pri_pkg.conf`.

Example

```
SERVICE_NAME[0]=″VVR″
SERVICE_CMD[0]="/opt/VRTSvrmcsg/vvr.sh monitor $[VXVM_DG[@]}"
SERVICE_RESTART[0]=-r "3"
```

❖ Modifying the `activate_disk_group` function

Add the following lines starting with the comments at the end of the `activate_disk_group` function.

```
function activate_disk_group
{
  :
  :
  :
  :


  for I in ${VXVM_DG[@]}
  do
    print "$(date '+%b %e %X') - \"$(hostname)\": Importing disk group $I."

    # If the disk group is already imported on this node,
    # check_dg will return 10. Then we can skip the vxdg import
    # since that would fail anyway.
    check_dg $I
    retval=$?
    if (( retval > 0 )) && (( retval != 10 )); then
    let 0
    test_return 22
    fi
    if (( retval != 10 )); then
    vxdg -tfC import $I
    fi
    test_return 23

    DiskGroup=$I

    eval $VXRVG              # Add this line for VVR recovery
    test_return 24           # Add this line

    eval $VXVOL
    test_return 24
  done
```

❖ Modifying the `deactivate_disk_group` Function

At the beginning of the `deactivate_disk_group` function add the following lines starting with the comments.

```
function deactivate_disk_group
{

/opt/VRTSvrmcsg/vvr.sh halt ${VXVM_DG[@]}# Add this line to stop RVG(s)
test_return 25                           # Add this line

for I in ${CVM_DG[@]}
do
.............
.............

}
```

3. To make the pri_pkg.cntl file available on host seattle2, copy the pri_pkg.cntl file to seattle2:

    ```
    # rcp /etc/cmcluster/pri_pkg/pri_pkg.cntl \
        seattle2:/etc/cmcluster/pri_pkg/pri_pkg.cntl
    ```

    Note Create the /etc/cmcluster/pri_pkg directory on seattle2 if it does not exist.

4. Create mount points /hr_mount01 and /hr_mount02 on seattle2.

Creating and Modifying the Package Control Script for the Secondary Cluster

Perform the following steps on the node `london1`:

1. Create the HP Serviceguard package control script by using the following command:

   ```
   cmmakepkg -s /etc/cmcluster/sec_pkg/sec_pkg.cntl
   ```

   You can also do this using SAM.

   The control script that is generated by the `cmmakepkg` command is created with the parameters set to default values. Customize this script to suit your requirements. Edit the package control script as indicated by the comments in that file.

2. Edit the package control script, `sec_pkg.cntl` using `vi` or any other editor.

   After the file is open edit the parameters. Refer to the following points for a description of some of the parameters that you need to change in the control script to configure HP Serviceguard for your environment.

   A sample package control file is present in the `/opt/VRTSvrmcsg/samples/london` directory.

❖ Specifying Disk Group Information

The `VXVM_DG` parameter is used to specify the name of the disk groups used by VVR.

Example

```
VxVM_DG[0]=”hrdg”
```

❖ Specifying File System Information

Do not specify any information here as the volumes under the Secondary RVG do not need to be mounted.

❖ Specifying the VVR Recovery Parameter

Use this parameter to specify the RVG recovery procedure to be used if the RVG fails at any point in time.

Specify the RVG recovery statement just before the volume recovery section in the script.

Example

```
VXRVG="/opt/VRTSvrmcsg/vvr.sh activate \$DiskGroup"
```

❖ Specifying the IP and Subnet

Use these parameters to specify the IP and subnet address of the host `london1`. Note that the IP that you specify must be the virtual IP used by the RLINK.

```
IP[0]="192.2.40.53"
SUBNET[0]="192.2.40.0"
```

- Specifying the service name information

  The service name must be the same as defined in the HP Serviceguard configuration file `sec_pkg.conf`.

  Example

```
SERVICE_NAME[0]="VVR"
SERVICE_CMD[0]="/opt/VRTSvrmcsg/vvr.sh monitor ${VxVM_DG[@]}"
SERVICE_RESTART[0]=-r "3"
```

- Modifying the `activate_disk_group` function

  Add the following lines starting with the comments at the end of the `activate_disk_group` function.

```
function activate_disk_group
{
:
:
:
for I in ${VXVM_DG[@]}
do
    print "$(date '+%b %e %X') - \"$(hostname)\": Importing disk group $I."

    # If the disk group is already imported on this node,
    # check_dg will return 10. Then we can skip the vxdg import
    # since that would fail anyway.
    check_dg $I
    retval=$?
    if (( retval > 0 )) && (( retval != 10 )); then
    let 0
    test_return 22
    fi
    if (( retval != 10 )); then
    vxdg -tfC import $I
    fi
    test_return 23

    DiskGroup=$I

    eval $VXRVG              # Add this line for VVR recovery
    test_return 24           # Add this line

    eval $VXVOL
    test_return 24
  done
```

- Modifying the deactivate_disk_group function

  Add the following lines starting with the comments at the beginning of the deactivate_disk_group function.

  ```
  function deactivate_disk_group
  {

  /opt/VRTSvrmcsg/vvr.sh halt ${VXVM_DG[@]}# Add this line to stop RVG(s)
  test_return 25                             # Add this line

  for I in ${CVM_DG[@]}
  do
  .............
  .............

  }
  ```

3. To make the sec_pkg.cntl file available on host london2, copy the sec_pkg.cntl file to london2:

   ```
   # rcp /etc/cmcluster/sec_pkg/sec_pkg.cntl
       london2:/etc/cmcluster/sec_pkg/sec_pkg.cntl
   ```

   Note Create the /etc/cmcluster/sec_pkg directory on london2 if it does not exist.

Proceed to the following section to start the package.

Starting the Package

Follow the steps that are provided below to start the packages on the Primary and Secondary clusters.

▼ To start the package on the Primary cluster

Perform the following steps on node `seattle1`:

1. Verify the package configuration.

   ```
   # cmcheckconf -v -P /etc/cmcluster/pri_pkg/pri_pkg.conf
   ```

2. Apply the package configuration.

   ```
   # cmapplyconf -v -P /etc/cmcluster/pri_pkg/pri_pkg.conf
   ```

3. Start the package.

   ```
   # cmrunpkg -n seattle1 pri_pkg
   ```

4. Check the package status.

   ```
   # cmviewcl -v
   ```

5. Enable the package to run on node `seattle2`.

   ```
   # cmmodpkg -e -n seattle2 pri_pkg
   ```

▼ To start the package on the Secondary cluster

Perform the following steps on the node `london1`:

1. Verify the package configuration.

   ```
   # cmcheckconf -v -P /etc/cmcluster/sec_pkg/sec_pkg.conf
   ```

2. Create the package by applying the package configuration.

   ```
   # cmapplyconf -v -P /etc/cmcluster/sec_pkg/sec_pkg.conf
   ```

3. Start the package.

   ```
   # cmrunpkg -n london1 sec_pkg
   ```

4. Check the package status.

   ```
   # cmviewcl -v
   ```

5. Enable the package to run on node london2.

   ```
   # cmmodpkg -e -n london2 sec_pkg
   ```

Testing the Package Switching and Failover 5

This chapter describes the procedure to test the package switching and failover between nodes in the cluster.

Testing the Package Switching Between Nodes

Follow the steps as specified, to test the package switching and failover.

For the Primary cluster

Perform the following steps on node `seattle1`:

1. Halt the package.

   ```
   # cmhaltpkg pri_pkg
   ```

2. Start the package on `seattle2`.

   ```
   # cmrunpkg -n seattle2 pri_pkg
   ```

3. Enable the package to run on node `seattle1`.

   ```
   # cmmodpkg -e -n seattle1 pri_pkg
   ```

4. Check the package status.

   ```
   # cmviewcl -v
   ```

5. Switch the package back to node `seattle1`.

   ```
   # cmhaltpkg pri_pkg
   # cmrunpkg -n seattle1 pri_pkg
   # cmmodpkg -e -n seattle2 pri_pkg
   # cmviewcl -v
   ```

▼ For the Secondary cluster

Perform the following steps on node `london1`:

1. Halt the package.

   ```
   # cmhaltpkg sec_pkg
   ```

2. Start the package on `london2`.

   ```
   # cmrunpkg -n london2 sec_pkg
   ```

3. Enable the package to run on node `london1`

   ```
   # cmmodpkg -e -n london1 sec_pkg
   ```

4. Check the package status.

   ```
   # cmviewcl -v
   ```

5. Switch the package back to node `london1`

   ```
   # cmhaltpkg sec_pkg
   # cmrunpkg -n london1 sec_pkg
   # cmmodpkg -e -n london2 sec_pkg
   # cmviewcl -v
   ```

Testing the Package Failover Between Nodes

▼ For the Primary cluster

Run the following commands on node `seattle1`

1. Unmount the volumes under the primary RVG.

```
# umount /hr_mount01
# umount /hr_mount02
```

2. Stop the primary RVG.

```
# vxrvg -g hrdg stop hr_rvg
```

The package `pri_pkg` should failover to the node `seattle2`.

▼ For the Secondary cluster

Run the following command on node `london1` .

Stop the secondary RVG.

```
# vxrvg -g hrdg stop hr_rvg
```

The package `sec_pkg` should failover to the node `london2`.