VERITAS™

VERITAS Cluster Server Enterprise Agent 4.1.01 for EMC MirrorView

Installation and Configuration Guide

AIX, HP-UX, Linux, Solaris

October 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

Contents

Preface

This document describes how to install and configure the VERITAS Cluster Server (VCS) enterprise agent for EMC MirrorView.

If this document is dated more than six months prior to the date you are installing the enterprise agent, contact VERITAS Technical Support to confirm you have the latest supported versions of the application and operating system.

To properly set up and configure your arrays, refer to the EMC Navisphere and the MirrorView Configuration and Installation Guide.

How This Guide is Organized

- Chapter 1. "Introduction" on page 1 introduces the VCS enterprise agent for EMC MirrorView and describes the tasks performed by the agent.
- Chapter 2. "Installing the MirrorView Agent" on page 7 provides instructions on installing the EMC MirrorView agent.
- Chapter 3. "Configuring the MirrorView Agent" on page 11 describes key configuration concepts and provides instructions on configuring the agent.
- Chapter 4. "Managing and Testing Clustering Support for MirrorView" on page 21 provides test scenarios and expected outcomes. It also describes how to remove the agent.

VERITAS Cluster Server Documentation

The following documents, along with the online help and the Release Notes, comprise the VCS documentation for this release:

Title	File Name
VERITAS Cluster Server Installation Guide	`vcs_install.pdf`
VERITAS Cluster Server User's Guide	`vcs_users.pdf`
VERITAS Cluster Server Bundled Agents Reference Guide	`vcs_bundled_agents.pdf`
VERITAS Cluster Server Agent Developer's Guide	`vcs_agent_dev.pdf`

See the Release Notes for a complete list of documents and VCS enterprise agent guides.

Conventions

The following conventions apply throughout the documentation set.

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

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 clusteringdocs@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

The VCS enterprise agent for EMC MirrorView provides application failover support and recovery in environments employing MirrorView to replicate data between EMC CLARiiON arrays.

About the MirrorView Agent

The VCS enterprise agent for EMC MirrorView monitors and manages the state of replicated CLARiiON LUNs attached to VCS nodes. The agent ensures that the system on which the MirrorView resource is online has safe exclusive access to the configured devices.

The agent can be used in single VCS replicated data clusters and multi-cluster environments set up using the VCS Global Cluster Option. The agent supports MirrorView in the synchronous mode and supports Individual mirrors or Consistent groups in asynchronous mode.

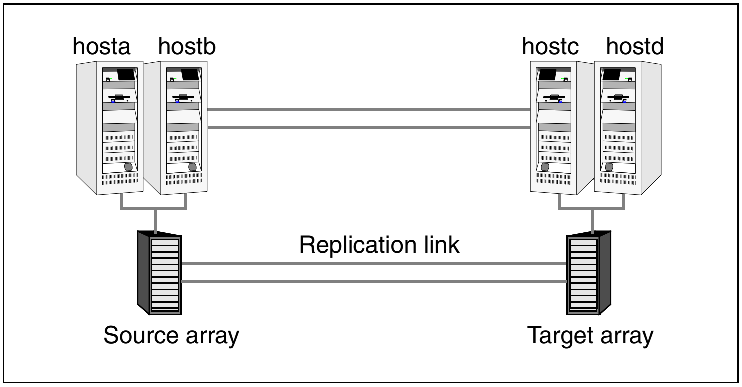
Supported Software and Hardware

The EMC MirrorView agent supports VCS 4.0 and VCS 4.1.

The agent supports versions of NaviCLI and FLARE code on CLARiiON found in the EMC hardware compatibility list.

Typical Setup

Clustering in a MirrorView environment typically consists of the following hardware infrastructure:



- ✔ The source array, comprising one or more hosts directly attached via SCSI or Fibre Channel to a CLARiiON array containing the mirror that is the primary image.
- ✔ The target array, comprising one or more hosts directly attached via SCSI or Fibre Channel to a CLARiiON array containing the mirror that is the secondary image. The secondary image LUNs are paired with the mirrored LUNs in the source array. These target hosts and the array must be at a significant distance from the source side to survive a disaster that may occur at the source side.
- ✔ Network heartbeats, LLT or TCP/IP, between the two data centers to determine their health. "Cluster Heartbeats" on page 16 for more information.
- ✔ In a replicated data cluster environment, all hosts are part of the same cluster. You must connect them with dual, dedicated networks that support LLT.
- ✔ In a global cluster environment, you must attach all hosts in a cluster to the same CLARiiON array.

Agent Operations

Configure EMC MirrorView with synchronous or asynchronous mode for replication. In asynchronous mode, you can replicate either individual LUNs or replicate consistency groups. MirrorView can also replicate LUNs or metaLUNs. The VERITAS Cluster Server Enterprise Agent for EMC MirrorView supports these configurations.

online

Synchronous

If the state of all local mirrors is MIRRORED, the agent creates a lock file on the local host to indicate that the resource is online. This makes the mirrors available for the application to use.

If one or more mirrors are not in the MIRRORED state, the agent runs a NaviCLI command to bring them into the MIRRORED state, which enables the application to use them.

- For secondary images in the SYNCHRONIZED state, the agent runs the `mirror -sync -promoteimage` command to promote the remote mirror. This also converts the current primary to secondary.
- For secondary images in the CONSISTENT state, the agent waits to check if the image has transitioned to the SYNCHRONIZED state.

  If the images have transitioned to the SYNCHRONIZED state, the agent then runs the `mirror -sync -promoteimage` command to promote the remote mirror. This also converts the current primary to secondary.

  If the image has not transitioned to the SYNCHRONIZED state, the agent checks if the remote array is accessible. If the remote array is accessible, then this condition indicates link failure—the image would be in a fractured condition.

  In case of fracture:

  - If the SplitTakeover attribute is set to 1, the agent forcibly promotes the secondary image.
  - If the SplitTakeover attribute is set to 0, the agent does not try to forcibly promote the secondary image, and becomes the administrator’s decision.

Asynchronous

Consistency groups

If the state of the group is SYNCHRONIZED, the agent creates a lock file on the local host to indicate that the resource is online. This makes the LUNs available for the application to use.

If one or more mirrors are not in the MIRRORED state, the agent checks to see if the remote array is accessible.

- If the remote array is not accessible, then this the agent checks the value of the attribute SplitTakeover before proceeding with any further actions.
- If the SplitTakeover attribute is set to 1, the agent forcibly promotes the secondary image.
- If the SplitTakeover attribute is set to 0, the agent does not try to forcibly promote the secondary image, and becomes the administrator's decision.
- If the remote array is accessible, then the agent runs the `mirror -async -promotegroup` command to promote the remote group.
- In case of a successful promotegroup operation, this also converts the current primary to secondary.

  If the promotegroup operation is not successful, then the agent initiates a synchronization.

  The agent periodically checks if the group is SYNCHRONIZED. After a successful synchronization, the agent promotes the group using the `mirror -async -promotegroup` command. If the synchronization is not successful, the agent times out.

Mirrors

If the state of all local mirrors is MIRRORED, the agent creates a lock file on the local host to indicate that the resource is online. This makes the mirrors available for the application to use.

If one or more mirrors are not in the MIRRORED state, the agent checks to see if the remote array is accessible.

- If the remote array is not accessible, then the agent checks the value of the attribute SplitTakeover before proceeding with any further actions.
- If the SplitTakeover attribute is set to 1, the agent forcibly promotes the secondary image.
- If the SplitTakeover attribute is set to 0, the agent does not try to forcibly promote the secondary image, and becomes the administrator's decision.

- If the remote array is accessible, then the agent runs the `mirror -async -promoteimage` command to promote the remote mirrors.
- In case of a successful promoteimage operation, this also converts the current primary to secondary.

If the promoteimage operation is not successful, then the agent initiates a synchronization.

The agent periodically checks if the group is SYNCHRONIZED. After a successful synchronization, the agent promotes the secondary mirror using the `mirror -async -promoteimage` command. If the synchronization is not successful, the agent times out.

monitor

Verifies that the lock file exists. If the lock file exists, the monitor entry point reports the status of the resource as online. If the lock file does not exist, the monitor entry point reports the status of the resource as offline.

open

Removes the lock file on the host where the entry point is called. This prevents potential concurrency violation if the service group fails over to another node.

Note The agent does not remove the lock file if the agent was started after an `hastop -force` command.

clean

Removes the lock file.

info

Gives the information about the mirrors (in case of synchronous mode of replication) and gives information about the mirrors/group in case of asynchronous mode of replication. It uses the `-sync listsyncprogress` and `-async -list` or `-async listgroups` commands to get this information.

resynch

Performs a resynchronization action.

Installing the MirrorView Agent 2

You must install the MirrorView enterprise agent on each node in the cluster. In global cluster environments, install the agent on each node in each cluster.

▼ To install the agent on AIX systems

1. Determine the device access name of the CD drive:

   ```
   # cd /dev
   # lsdev -C -c cdrom
   ```

   The output resembles:

   ```
   cd0 Available 10-60-00-4,0 16 Bit SCSI Multimedia CD-ROM Drive
   ```

   In this example, the CD device access name is `cd0`.

2. Insert the CD into a drive connected to the system.

3. Mount the CD:

   ```
   # mkdir -p /cdrom
   # mount -V cdrfs -o ro /dev/cd0 /cdrom
   ```

4. Add the filesets for the software:

   ```
   # cd /cdrom
   # installp -ac -d /cdrom/aix/replication/mirrorview_agent
   /4.1.01/pkgs/VRTSvcsm.rte.bff VRTSvcsm
   ```

▼ To install the agent on HP-UX systems

1. Insert the disc into a drive connected to the host.

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

3. Determine the block device file for the disc drive:

   ```
   # ioscan -fnC disk
   ```

   For example, the listing may indicate the block device is `/dev/dsk/c1t2d0`.

4. Start the Portable File System (PFS).

   ```
   # nohup pfs_mountd &
   # nohup pfsd &
   ```

5. Mount the disc:

   ```
   # /usr/sbin/pfs_mount -t rrip /dev/dsk/c#t#d# /cdrom
   ```

   The variable `/c#t#d#` represents the location of the drive.

6. Install the agent software. Type one of the following commands depending on the operating system on the node.

   For HP-UX (PA) type:

   ```
   # swinstall -s /cdrom/hpux/replication/mirrorview_agent/4.1.01/PA/
      depot/VRTSvcsm
   ```

   For HP-UX (IA) type:

   ```
   # swinstall -s /cdrom/hpux/replication/mirrorview_agent/4.1.01/IA/
      depot/VRTSvcsm
   ```

▼ To install the agent on Linux systems

1. Log in as root.

2. Insert the software disc into a drive connected to the system and mount the disc.

   ```
   # mount -o ro /dev/cdrom /mnt/cdrom
   ```

3. Navigate to the /mnt directory:

   ```
   # cd /mnt/cdrom
   ```

4. Install the agent software. Type one of the following commands depending on the operating system on the node. The Linux packages are located at: `/linux/replication/mirrorview_agent/4.1.01/rpms/`

   For Red Hat Enterprise Linux 4 (RHEL 4) on i686:

   ```
   # rpm -ivh rpms/VRTSvcsm-4.1.01.0-GA_RHEL4.i686.rpm
   ```

   For Red Hat Enterprise Linux 4 (RHEL 4) on IA64:

   ```
   # rpm -ivh rpms/VRTSvcsm-4.1.01.0-GA_RHEL4.i686.rpm
   ```

   For Red Hat Enterprise Linux 4 (RHEL 4) x86_64:

   ```
   # rpm -ivh rpms/VRTSvcsm-4.1.01.0-GA_RHEL4.i686.rpm
   ```

   For SUSE Linux Enterprise Server 9 (SLES 9) i586:

   ```
   # rpm -ivh rpms/VRTSvcsm-4.1.01.0-GA_SLES9.i586.rpm
   ```

   For SUSE Linux Enterprise Server 9 (SLES 9) on IA64:

   ```
   # rpm -ivh rpms/VRTSvcsm-4.1.01.0-GA_SLES9.i586.rpm
   ```

   For SUSE Linux Enterprise Server 9 (SLES 9) x86_64:

   ```
   # rpm -ivh rpms/VRTSvcsm-4.1.01.0-GA_SLES9.i586.rpm
   ```

▼ To install the agent on Solaris systems

1. Create a temporary directory for installation:

   ```
   # mkdir /tmp/install
   ```

2. Insert the disc into a drive connected to the system. Type the command:

   ```
   # cd /cdrom/cdrom0
   ```

3. Copy the compressed package files from the software disc to the temporary directory:

   ```
   # cp -r solaris/replication/mirrorview_agent/4.1.01/pkgs/VRTSvcsm
    /tmp/install
   ```

4. Go to the temporary directory and unzip the compressed package files:

   ```
   # cd /tmp/install
   # gunzip VRTSvcsm.gz
   ```

5. Extract the compressed file from the tar file:

   ```
   # tar -xvf VRTSvcsm.tar
   ```

6. Install the agent and wizard package:

```
# pkgadd -d . VRTSvcsm
```

Configuring the MirrorView Agent 3

Most applications configured in VCS can be adapted to a disaster recovery environment by:

- Converting their LUNs to CLARiiON LUNs
- Synchronizing the mirrors
- Adding the EMC MirrorView agent to the service group

Before Configuring the Agent

- ✔ Verify that the agent is installed on all systems in the cluster.
- ✔ Verify that the hardware setup for the agent. See "Typical Setup" on page 2 for more information.
- ✔ Make sure the cluster has an effective heartbeat mechanism in place. See "Cluster Heartbeats" on page 16 for more information.
- ✔ Review the agent's resource type definition and its attribute definitions.
- ✔ Review the section "Configuration Concepts" on page 16. This section presents information about how VCS behaves during failover and how you can set attributes to customize VCS behavior.

Resource Type Definition

```
type MirrorView (
        static keylist SupportedActions = { resync }
        static int MonitorInterval = 300
        static int NumThreads = 1
        static int OfflineMonitorInterval = 0
        static int RestartLimit = 1
        static str ArgList[] = { NaviCliHome, LocalArraySPNames,
        RemoteArraySPNames, Mode, GrpName, MirNames, SplitTakeover }
        str NaviCliHome = "/opt/Navisphere/bin"
        str LocalArraySPNames[]
        str RemoteArraySPNames[]
        str Mode
        str GrpName
        str MirNames[]
        int SplitTakeover
        temp str VCSResLock
)
```

Attribute Definitions

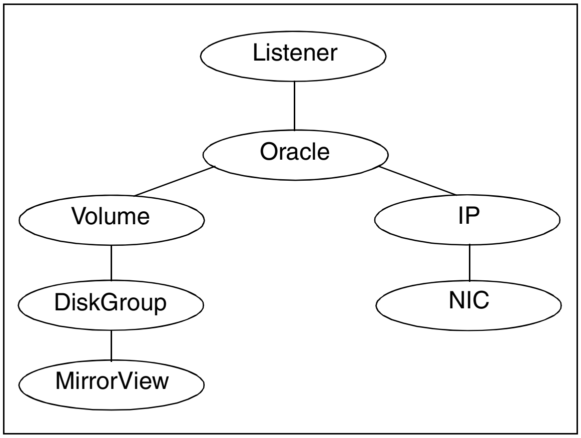
Required attribute	Description
NaviCliHome	NaviCLI installation directory Type and dimension: string-scalar
LocalArraySPNames	The list of storage processors within the array to which the local hosts are connected. Can be names or IP addresses. Type and dimension: string-vector
RemoteArraySPNames	The list of storage processors within the array to which the remote hosts are connected. Can be names or IP addresses. Type and dimension: string-vector
RemoteArrayName	The name of the remote CLARiiON array to which the remote hosts are connected. Type and dimension: string-vector
Mode	The replication mode, which is either: `sync` or `async`. Type and dimension: string-scalar
GrpName	The name of the consistency group to which the mirrors belong. This applies only if the mode is `async`. Type and dimension: string-scalar
MirNames	This is a list of individual mirrors that are a part of the replication relationship and managed by VCS. This attribute is ignored if you specify the GrpName attribute. Type and dimension: string-vector

Required attribute	Description
SplitTakeover	This integer indicates whether VCS should forcefully promote a secondary to a primary. In case of a link-failure between the two arrays, the state of the mirror remains consistent or out-of-sync. Under such circumstances, if the application has to failover—due to disaster or user-driven action—mirrors are not in a SYNCHRONIZED state. If the value of the SplitTakeOver attribute is 1, the agent fails over when it discovers link failures and determines that mirrors are out of sync. If the value of the attribute is 0, agent does not fail over and the administrator must to determine what to do. Type and dimension: integer-scalar

Internal attribute	Description
VCSResLock	This attribute is used by the agent to guarantee serialized management in case of a parallel application. Do not modify this value. Type and dimension: temporary-string

Sample Configuration

This dependency graph shows a VCS service group that has a resource of type MirrorView. The DiskGroup resource depends on the MirrorView resource.



Asynchronous Mode

You can configure a resource of type MirrorView in the `main.cf` file. In this example, the resource is configured for asynchronous mode and consistency groups.

```
MirrorView mir (
    NaviCliHome = "/opt/Navisphere/bin"
    LocalArraySPNames @sys1= = { "Local_SP1_Name", "Local_SP1_IP" }
    LocalArraySPNames @sys2 = { "Local_SP2_Name", "Local_SP2_IP" }
    RemoteArraySPNames @sys1 = { "Local_SP2_IP", "Remote_SP2_Name" }
    RemoteArraySPNames @sys2= { "Local_SP1_IP", "Remote_SP1_Name" }

    Mode = async
    GrpName = consistency_grp1

    SplitTakeover = 0
    )
```

In the previous example, if you want set up asynchronous replication with individual mirrors—no consistency groups—replace the lines beginning with Mode and GrpName with the following lines:

```
    Mode = async
    MirNames = { "async_mir1", "async_mir2" }
    GrpName = ""
```

In the following example, if you want to configure the resource for synch mode and specify the mirror names, replace the lines beginning with Mode and GrpName with the following lines:

```
Mode = sync
MirNames = { "sync_mir1", "sync_mir2" }
GrpName = ""
```

Cluster Heartbeats

In a replicated data cluster, robust heartbeating is accomplished through dual, dedicated networks over which the Low Latency Transport (LLT) runs. Additionally, you can configure a low-priority heartbeat across public networks.

In a global cluster, network heartbeating is accomplished by sending ICMP pings over the public network between the two sites. VCS global clusters minimize the risk of split-brain by sending ICMP pings to highly available IP addresses and by notifying administrators when the sites cannot communicate.

If the loss of heartbeats occurs due to the failure of all hosts in the primary cluster, a failover may be required even if the array is alive. In any case, it is important to distinguish between a host-only crash and a complete site failure. In a host-only crash, only the ICMP heartbeat signals a failure via an SNMP trap. No cluster failure notification occurs because a surviving heartbeat exists. This trap is the only notification to fail over an application.

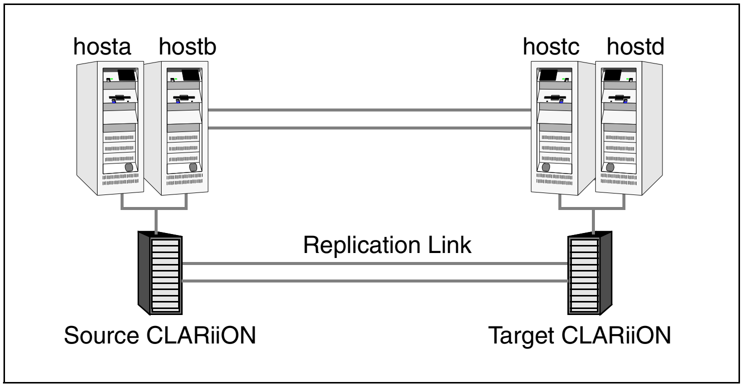
Configuration Concepts

This section describes some failure scenarios and provides guidelines on how to configure the agent.

Individual Component Failure

In a replicated data cluster, you can prevent unnecessary failover or failback by configuring hosts attached to an array as part of the same system zone. VCS attempts to fail over applications within the same system zone before failing them over across system zones.

In the following graphic, hosta and hostb are in one system zone, and hostc and hostd are in another system zone. The SystemZones attribute enables you to create these zones.



You can modify the SystemZones attribute using the following command:

```
# hagrp -modify grpname SystemZones hosta 0 hostb 0 hostc 1 hostd 1
```

The variable *grpname* represents the service group in the cluster.

This command creates two system zones: zone 0 with hosta and hostb, zone 1 with hostc and hostd.

Global clusters do not require system zones because failover occurs on a remote cluster if all local targets have been exhausted.

As long as a secondary image is available, MirrorView sends the writes to the secondary image immediately in synchronous mode. It does so periodically in asynchronous mode.

If the period is too long, you can perform synchronization using the resync action. The supported resync action is defined in the MirrorView resource type.

All Host or All Application Failure

If all hosts on the source CLARiiON are disabled or if the application cannot start successfully on any source hosts, but both arrays are operational, the service group fails over.

In replicated data cluster environments, the failover can be automatic, whereas in global cluster environments, failover requires user confirmation by default. In both environments, multiple service groups can fail over in parallel.

Total Site Disaster

In a total site failure, all hosts and the array are completely disabled, either temporarily or permanently.

In a replicated data cluster, site failure is detected the same way as a total host failure, that is, the loss of all LLT heartbeats.

In a global cluster, VCS detects site failure by the loss of all configured heartbeats.

A total disaster renders the devices on the surviving array in the FRACTURED state. If the SplitTakeover attribute is set to its default value of 1, the online entry point runs the ‘promote’ operation. If the attribute is set to 0, no takeover occurs and the online entry point times out and faults.

The online entry point detects whether any synchronization was in progress when the source array was lost. Since the target devices are inconsistent until the synchronization completes, the agent does not write-enable the devices, but it times out and faults. You must restore consistent data from a snapshot or tape backup.

Replication Link Failure

If the two arrays are healthy and the link fails and is restored, and if a failover is initiated while one or more devices are in the SYNCHRONIZING state, the MirrorView waits for the synchronization to complete before running the promote operation. If the agent times out before the synchronization completes, the resource faults.

If the SplitTakeover attribute is set to 0, the agent does not attempt a promote operation, but it times out and faults. If you write-enable the devices manually, the agent can come online after it is cleared.

Split-brain

Split-brain occurs when all heartbeat links between the primary and secondary hosts are cut and each side mistakenly thinks the other side is down. To minimize the effects of split-brain, it is best if the cluster heartbeat links pass through similar physical infrastructure as the replication links so that if one breaks, so does the other.

In a replicated data cluster, VCS attempts to start the application assuming a total disaster because the source and array are unreachable. Once the heartbeats are restored, VCS stops the applications on one side and restarts the VCS engine (HAD) to eliminate concurrency violation of the same group being online at two places simultaneously. You must resynchronize the volumes manually using the `symrdf merge` or `symrdf restore` commands.

In a global cluster, you can confirm the failure before failing over the service groups. You can check with the site administrator to identify the cause of the failure. If you do mistakenly fail over, the situation is similar to the replicated data cluster case; however, when the heartbeat is restored, VCS does not stop HAD at either site. VCS forces you to choose which group to take offline. You must resynchronize data manually.

If it is physically impossible to place the heartbeats alongside the replication links, there is a possibility that the cluster heartbeats are disabled, but the replication link is not. A failover transitions the original target volumes to source volumes and vice-versa. In this case, the application faults because its underlying volumes become write-disabled. VCS tries to fail the application over to another host, causing the same consequence in the reverse direction. This phenomenon continues until the group comes online on the final node. This situation can be avoided by setting up your infrastructure such that the loss of heartbeat links also means the loss of replication links.

Configuring the Agent Manually

This section describes agent configuration.

Configuring the Agent in a Global Cluster

▼ **To configure the agent in a global cluster**

1. If the agent resource type (mirrorview) is not added to your configuration, add it.

    a. Start Cluster Manager and log on to the cluster.

    b. From the Cluster Explorer **File** menu, choose **Import Types** and select `/etc/VRTSvcs/conf/mirrorview.cf.`

    c. Click **Import**.

    d. Save the configuration.

2. Perform the following tasks for each service group in each cluster that uses replicated data:

    a. Add a resource of type mirrorview at the bottom of the service group.

    b. Configure the attributes of the resource.

c. If the service group is not configured as a global group, configure the service group using the Global Group Configuration Wizard. See the *VERITAS Cluster Server User's Guide* for more information.

d. Change the ClusterFailOverPolicy from the default, if necessary. VERITAS recommends keeping the default, which is Manual, to minimize the chance of failing over on a split-brain.

Configuring the Agent in a Replicated Data Cluster

▼ **To configure the agent in a replicated data cluster**

1. If the agent resource type (mirrorview) is not added to your configuration, add it.

   a. Start Cluster Manager and log on to the cluster.

   b. From the Cluster Explorer **File** menu, choose **Import Types** and select `/etc/VRTSvcs/conf/mirrorview.cf.`

   c. Click **Import**.

   d. Save the configuration.

2. Perform the following tasks for each service group that uses replicated data:

   a. Add a resource of type mirrorview at the bottom of the service group.

   b. Configure the attributes of the mirrorview resource. Note that some attributes must be localized to reflect values for hosts attached to different arrays.

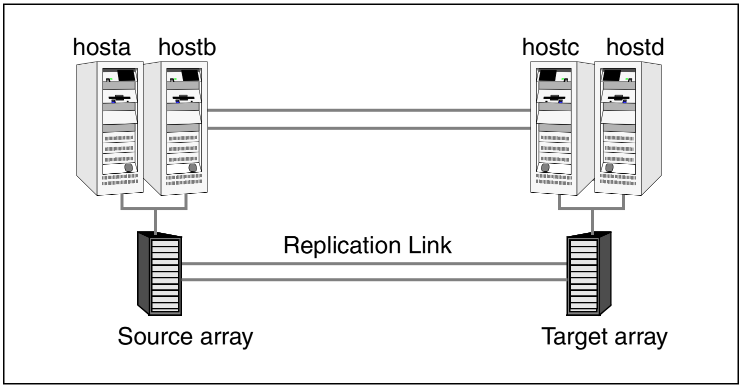
   c. Set the SystemZones attribute for the service group to reflect which hosts are attached to the same array.

Managing and Testing Clustering Support for MirrorView 4

After configuring the MirrorView agent in a VCS environment, you can perform some basic tests to verify the implementation. This chapter describes some test scenarios and expected behavior.

These tests assume the following environment:

- Two hosts (hosta and hostb) are attached to the source CLARiiON array, and the other hosts are attached to the target CLARiiON array. The application is running on hosta and devices in the local array are read-write enabled, in the SYNCHRONIZED state.
- A replicated data cluster has two dedicated heartbeat links; a global cluster has one network heartbeat. The test scenario is similar for both environments.



Service Group Migration

Verify the service group can migrate to different hosts in the cluster.

▼ To perform the service group migration test

1. Migrate the service group to a host attached to the same array.

   a. In the **Service Groups** tab of the Cluster Explorer configuration tree, right-click the service group.

   b. Click **Switch To**, and click the system attached to the same array (hostb) from the menu.

      The service group comes online on hostb and local image remains in the MIRRORED state.

2. Migrate the service group to a host attached to a different array:

   a. In the **Service Groups** tab of the Cluster Explorer configuration tree, right-click the service group.

   b. Click **Switch To**, and click the system attached to the another array (hostc) from the menu.

      The service group comes online on hostc and the role of the images there transition to primary.

3. Accumulate dirty tracks on the new source-side and update them back on the target:

   ```
   hares -action mirrorview_res_name resync -sys hostc
   ```

   The variable *mirrorview_res_name* represents the name of the MirrorView resource.

4. After the devices transition to a source SYNCHRONIZED state, migrate the service group back to its original host:

   a. In the **Service Groups** tab of the Cluster Explorer configuration tree, right-click the service group.

   b. Click **Switch To**, and click the system on which the group was initially online (hosta).

The group comes online on hosta. The devices return to the RW/SYNCINPROG state at the array attached to hosta and hostb, and then eventually transition to the SYNCHRONIZED state.

Host Failure

In this scenario, the host where the application runs is lost. Eventually all the hosts in the system zone or cluster are lost.

▼ **To perform the host failure test**

1. Halt or shut down the host where the application runs.

   The service group fails over to hostb and devices are in the SYNCHRONIZING state.

2. Halt or shut down hostb.

   In a replicated data cluster, the group fails over to hostc or hostd depending on the FailOverPolicy in the cluster.

   In a global cluster, a cluster down alert appears and gives you the opportunity to fail over the service group manually.

   In both environments, the role of the devices changes from secondary to primary and starts on the target host.

3. Reboot the two hosts that were shut down.

4. Switch the service group to its original host when VCS starts.

   a. In the **Service Groups** tab of the Cluster Explorer configuration tree, right-click the service group.

   b. Click **Switch To**, and click the system on which the service group was initially online (hosta).

   The service group comes online on hosta and devices transition to the SYNCHRONIZING state and then to the SYNCHRONIZED state.

Disaster Test

Shut down all hosts on the source side and shut down the source array. If shutting down the source array is not feasible, change the value of the RemoteArraySPNames in the target side to non-existent names and IP addresses; this action mimics a disaster scenario from the target's point of view.

In a replicated data cluster, the service group fails over to hostc or hostd if all devices were originally SYNCHRONIZED—no synchronization was in progress at the time of disaster.

In a global cluster, the administrator is notified of the failure. The administrator can then initiate the failover by declaring an outage.

Failback Test

▼ To perform the failback test for asynchronous mode with Consistency groups

1. Remove all the mirrors form the consistency group on the old primary.
2. Destroy the consistency group on the old primary.
3. Forcefully destroy the remote mirrors on the old primary.
4. Remove the LUNs from the storage group on the old primary.
5. Remove the mirrors from the consistency group on the new primary.
6. Add secondary images to each of the remote mirrors on the new primary.
7. Add the mirrors into the consistency group on the new primary.
8. Add the LUNs, where the secondary image resides, into the appropriate storage group on the old primary

Between step 5 and step 7, the LUNs become vulnerable to data corruption, for example if one of the LUNs sustained hardware damage and failed.

During this window, since the mirrors are not a part of the consistency group, the writes to other mirrors that were a part of the consistency group are not stopped—which could result in data corruption.

▼ To perform the failback test for synchronous and asynchronous mode with Individual mirrors

1. Forcefully destroy the remote mirrors on the old primary.

2. Remove the LUNs from the storage group on the old primary.

3. Add secondary images to each of the remote mirrors on the new primary.

4. Add the LUNs, where the secondary image resides, into the appropriate storage group on the old primary.

In either of the above modes, the original contents of the old primary are lost.

Removing the Agent

▼ To remove the agent from AIX systems

Type the following command on each system to remove the agent. Answer prompts accordingly:

```
# installp -u VRTSvcsm
```

▼ To remove the agent from HP-UX systems

Type the following command on each system to remove the agent. Answer prompts accordingly:

```
# swremove VRTSvcsm
```

▼ To remove the agent from Linux systems

Type one of the following commands, depending on the operating system on the cluster node.

For Red Hat Enterprise Linux 4 (RHEL 4) on i686:

```
# rpm -e VRTSvcsm-4.1.01.0-GA_RHEL4.i686.rpm
```

For Red Hat Enterprise Linux 4 (RHEL 4) on IA64:

```
# rpm -e VRTSvcsm-4.1.01.0-GA_RHEL4.i686.rpm
```

For Red Hat Enterprise Linux 4 (RHEL 4) x86_64:

```
# rpm -e VRTSvcsm-4.1.01.0-GA_RHEL4.i686.rpm
```

For SUSE Linux Enterprise Server 9 (SLES 9) i586:

```
# rpm -e VRTSvcsm-4.1.01.0-GA_SLES9.i586.rpm
```

For SUSE Linux Enterprise Server 9 (SLES 9) on IA64:

```
# rpm -e VRTSvcsm-4.1.01.0-GA_SLES9.i586.rpm
```

For SUSE Linux Enterprise Server 9 (SLES 9) on x86_64:

```
# rpm -e VRTSvcsm-4.1.01.0-GA_SLES9.i586.rpm
```

Repeat these instructions on each node in the cluster.

▼ To remove the agent from Solaris systems

Type the following command on each system to remove the agent. Answer prompts accordingly:

```
# pkgrm VRTSvcsm
```

Index