Veritas™ Cluster Server Hardware Replication Agent for EMC MirrorView Configuration Guide

Windows 2000, Windows Server 2003

5.0

11293739

symantec.

Veritas Cluster Server Hardware Replication Agent for EMC MirrorView

Veritas Cluster Server 5.0

Symantec Corporation
20330 Stevens Creek Blvd.
Cupertino, CA 95014
www.symantec.com

Third-party legal notices

Third-party software may be recommended, distributed, embedded, or bundled with this Symantec product. Such third-party software is licensed separately by its copyright holder. All third-party copyrights associated with this product are listed in the accompanying release notes.

Windows is a registered trademark of Microsoft Corporation.

Licensing and registration

Veritas Cluster Server is a licensed product.

Technical support

For technical assistance, visit http://entsupport.symantec.com and select phone or email support. Use the Knowledge Base search feature to access resources such as TechNotes, product alerts, software downloads, hardware compatibility lists, and our customer email notification service.

Contents

Chapter 1

Introduction

This chapter contains the following topics:

About the EMC MirrorView agent

The VCS enterprise agent for EMC MirrorView provides application failover support and recovery in environments employing MirrorView to replicate data between EMC CLARiiON arrays. It monitors and manages the state of replicated CLARiiON LUNs attached to VCS nodes. The agent ensures that the system where the MirrorView resource runs has safe and exclusive access to the configured devices.

You can use the agent in replicated data clusters and global clusters. The agent supports MirrorView in the synchronous mode and supports individual mirrors or consistency groups in asynchronous mode.

Configure EMC MirrorView for use 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.

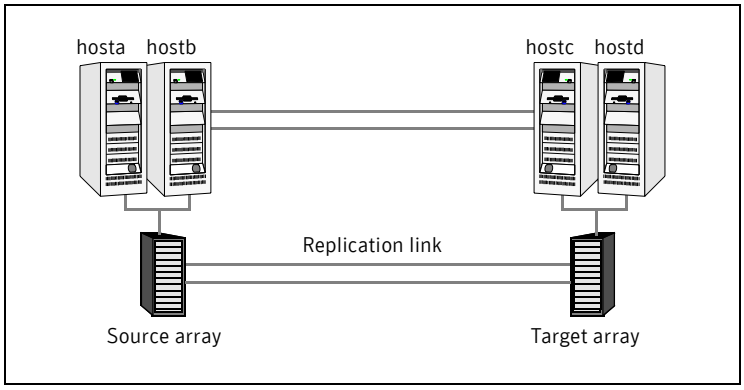
Supported software and hardware

The EMC MirrorView agent supports VCS 5.0.

To determine the supported versions of NaviCLI and FLARE code that are on CLARiiON arrays, consult the EMC hardware compatibility list.

Typical setup

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



- The source array consists of one or more hosts with direct connection to a CLARiiON array. The array contains the mirror that is the primary image and the direct connection uses either SCSI or Fibre Channel.
- The target array consists of one or more hosts with a direct connection to another CLARiiON array. The array contains the mirror that is the secondary image and the connection uses either SCSI or Fibre Channel.
  The secondary image LUNs pairs with the mirrored LUNs in the source array. The target hosts and the array must be at a significant distance from the source side to survive a source-side disaster
- Network heartbeats, LLT or TCP/IP, between the two data centers to determine their health.
  See “About cluster heartbeats” on page 23 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.

MirrorView agent functions

The VCS enterprise agent for EMC MirrorView monitors and manages the state of replicated CLARiiON LUNs attached to VCS nodes. Agent functions bring resources online, take them offline, and perform different monitoring actions. Agent functions are also known as entry points.

online — Creates a lock file on the local host. This lock indicates that the resource is online and makes the mirrors available for the application to use. The agent performs specific actions depending on the state of the mirrors.

See "About the MirrorView agent's online function" on page 11.

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 action 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 a `hastop<-all | -local> -force` command.

clean — Removes the lock file.

info — The info function gives the information about the mirrors (in case of synchronous mode of replication). It also 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.

action
- resync
  Performs a resynchronization action.
- addArrayUser
  Creates security for the SYSTEM user on the local and remote arrays.

About the MirrorView agent's online function

The agent online function performs specific actions depending on the state of the mirrors.

Synchronous state

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 state

You can configure the online function for either consistency groups or mirrors in asynchronous mode.

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 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, the operation 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.
- A successful promoteimage operation 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.

Chapter 2

Installing the EMC MirrorView agent

This chapter contains the following topics:

Before you install the MirrorView agent

Set up your cluster. For information about installing and configuring VCS, see the *Veritas Storage Foundation and High Availability Solutions Installation and Upgrade Guide*.

Set up replication and the required hardware infrastructure.

See "Typical setup" on page 9.

Installing the MirrorView agent

If you did not install the EMC MirrorView agent when you installed Veritas Storage Foundation for Windows High Availability (SFW HA), follow these instructions to install the agent.

You must install the MirrorView agent on each node in the cluster. In global cluster environments, install the agent on each node in each cluster. These instructions assume that you have already installed SFW HA.

To install the agent

1. Open the Windows Control Panel and click **Add or Remove Programs**.
2. Click the SFW HA Server Components entry and click **Change**.
3. On the installer screen, click **Add or Remove** and click **Next**.
4. In the Option Selection dialog box, select the agent and click **Next**.
5. The installer validates the system for installation.
   If a system is rejected, the Comments column displays the cause of rejection. Highlight the system to view detailed information about the failure in the Details box. Resolve the error, highlight the node in the selected systems list, and click **Validate Again**.
   After all the systems are accepted, click **Next**.
6. An informational message appears if you selected the DMP option. Review the information and click **OK** to continue.
7. Review the summary of your selections and click **Next**.
8. Click **Update** to start the installation.
9. The installer displays the status of installation. After the installation is complete, review the installation report and click **Next**.
10. Click **Finish**.

Removing the MirrorView agent

This section describes steps for uninstalling the agent. Do not attempt to remove the agent if service groups accessing the shared storage are online.

1. Open the Windows Control Panel and click **Add or Remove Programs**.
2. Click the SFW HA Server Components entry and click **Remove**.
3. Click **Next**.
4. In the Option Selection dialog box, select the MirrorView agent and click **Next**.
5. The installer validates the system for uninstallation.
   If a system is rejected, the Comments column displays the cause of rejection. Highlight the system to view detailed information about the failure in the Details box. Resolve the error, highlight the node in the selected systems list, and click **Validate Again**.
   After all the systems are accepted, click **Next**.
6. Review the summary of your selections and click **Uninstall**.
7. The installer displays the status of uninstallation.
8. After the uninstallation is complete, review the report and click **Next**.
9. Click **Finish**.

Note: For Win IA64 and Win x64 architectures, you will have to manually delete the agent directory if it is not removed after the uninstallation.

Chapter 3

Configuring the EMC MirrorView agent

This chapter contains the following topics:

Configuration concepts

Review the resource type definition and the attribute definitions for the agent.

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 = "C:\Program Files\EMC\Navisphere CLI"
        str LocalArraySPNames[]
        str RemoteArraySPNames[]
        str Mode
        str GrpName
        str MirNames
        int SplitTakeover
        temp str VCSResLock
)
```

Attribute definitions for the MirrorView agent

Review the description of the agent attributes.

Required attributes

You must assign values to required attributes

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
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 function applies only if the mode is `async`.

Type and dimension: string-scalar

MirNames

This function lists 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

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
- the agent determines that mirrors are out of sync

If the value of the attribute is 0, agent does not fail over and the administrator must determine what to do.

Type and dimension: integer-scalar

Internal attribute

Do not modify internal attributes.

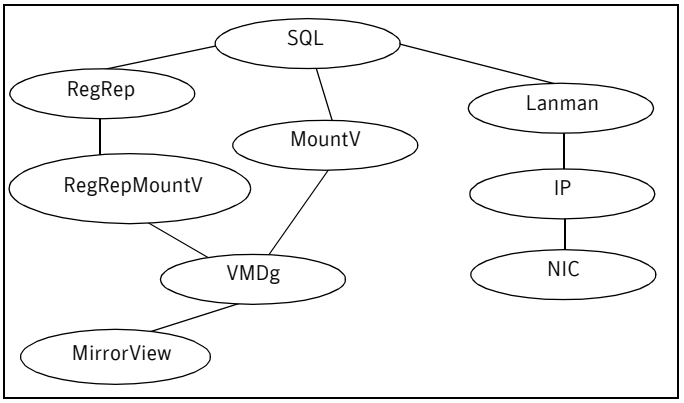
VCSResLock

This agent uses this attribute to guarantee serialized management in case of a parallel application. Do not modify this value.

Type and dimension: temporary-string

Sample configuration

The following 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 = "C:\Program Files\EMC\Navisphere CLI"
    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
    )
```

If you want to set up asynchronous replication with individual mirrors—no consistency groups—replace the lines beginning with Mode and GrpName in the `main.cf` file with the following lines:

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

If you want to configure the resource for synchronous mode and specify the mirror names, replace the lines beginning with Mode and GrpName in the `main.cf` file as follows:

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

)
```

Before you configure the MirrorView agent

Before you configure the agent:

- Review the configuration concepts, which describe the agent's type definition and attributes.
  See "Configuration concepts" on page 20.
- Verify that the agent is installed on all systems in the cluster.
- Verify the hardware setup for the agent.
  See "Typical setup" on page 9.
- Make sure the cluster has an effective heartbeat mechanism in place.
  See "About cluster heartbeats" on page 23.
- Set up system zones in replicated data clusters.
  See "About configuring system zones in replicated data clusters" on page 24.
- Set up an effective heartbeat mechanism to prevent split-brain.
  See "About preventing split-brain" on page 24.

About 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, VCS sends ICMP pings over the public network between the two sites for network heartbeating. To minimize the risk of split-brain, VCS sends ICMP pings to highly available IP addresses. VCS global clusters also notify the administrators when the sites cannot communicate.

Heartbeat loss may occur due to the failure of all hosts in the primary cluster. In such a scenario, a failover may be required even if the array is alive. In any case, a host-only crash and a complete site failure must be distinguished. In a host-only crash, only the ICMP heartbeat signals a failure by an SNMP trap. No cluster failure notification occurs because a surviving heartbeat exists. This trap is the only notification to fail over an application.

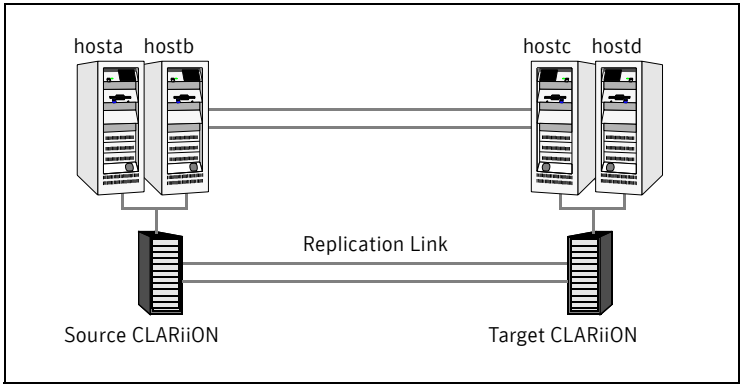
About preventing split-brain

Split-brain occurs when all heartbeat links between the primary and secondary hosts are cut. In this situation, each side mistakenly assumes that the other side is down. Minimize the effects of split-brain by ensuring the cluster heartbeat links pass through similar physical infrastructure as the replication links so that if one breaks, so does the other.

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 primary host to the secondary host and vice-versa. In this case, the application faults because its underlying volumes become write-disabled, causing the service group to fault. VCS tries to fail it over to another host, causing the same consequence in the reverse direction. This phenomenon continues until the group comes online on the final node. You can avoid this situation by setting up your infrastructure such that loss of heartbeat links also mean the loss of replication links.

About configuring system zones in replicated data clusters

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.



This example depicts a sample configuration where 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:

```
C:\> 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 host b, 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.

Configuring the agent

You can adapt most of the applications that you configure in VCS to a disaster recovery environment by:

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

Configuring the agent manually in a global cluster

Configuring the agent manually in a global cluster involves the following tasks.

To configure the agent in a global cluster

1. Start Cluster Manager and log on to the cluster.
2. If the agent resource type (mirrorview) is not added to your configuration, add it. From the Cluster Explorer **File** menu, choose **Import Types** and select C:\Program Files\Veritas\Cluster Server\conf\config\MirrorViewTypes.cf
3. Click **Import**.
4. Save the configuration.
5. Add a resource of type mirrorview at the bottom of the service group.
6. Configure the attributes of the mirrorview resource.

7 If the service group is not configured as a global group, configure the service group using the Global Group Configuration Wizard.

8 Change the ClusterFailOverPolicy from the default, if necessary. Symantec recommends keeping the default, which is Manual, to minimize the chance of failing over on a split-brain.
Repeat step 5 through step 8 for each service group in each cluster that uses replicated data.

Configuring the agent manually in a replicated data cluster

Configuring the agent manually in a replicated data cluster involves the following tasks.

To configure the agent in a replicated data cluster

1 Start Cluster Manager and log on to the cluster.

2 If the agent resource type (mirrorview) is not added to your configuration, add it. From the Cluster Explorer **File** menu, choose **Import Types** and select C:\Program Files\Veritas\Cluster Server\conf\config\MirrorViewTypes.cf.

3 Click **Import**.

4 Save the configuration.

5 In each service group that uses replicated data, add a resource of type mirrorview at the bottom of the service group.

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

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

Executing the addArrayUser action

Executing the addArrayUser action is a critical step in configuring the EMC MirrorView agent in both global clusters and replicated data clusters.

The MirrorView agent runs under the SYSTEM user context. You must execute the addArrayUser action to create the required security files, thus allowing you as a SYSTEM user to run commands on both local and remote arrays.

To execute the addArrayUser action

1. Right click the mirrorview resource and select **Actions...** .
2. From the list of actions, select **addArrayUser**.
3. Select the system on which to execute the action.
4. From the list of action arguments, click the plus button to add a new argument. Type in a non-blank password for the SYSTEM user to use.
5. Click **OK** to execute the action.
6. Review the results of the action and click **OK**.
7. Repeat these steps for all nodes that need to access the arrays.

Chapter 4

Managing and testing clustering support for EMC MirrorView

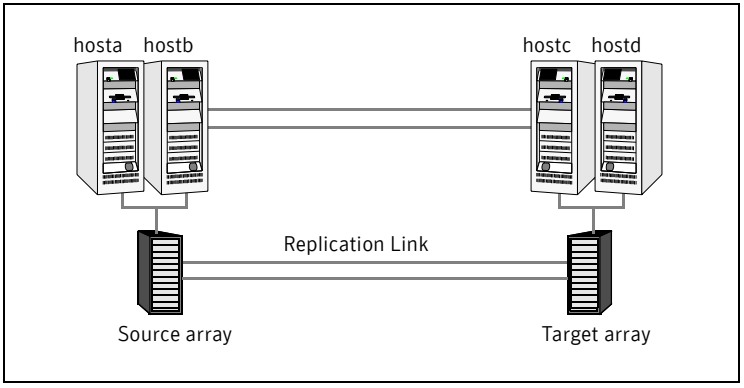
This chapter contains the following topics:

Typical test setup

A typical test environment includes:

- Two hosts (hosta and hostb) attached to the source CLARiiON array.
- Two hosts (hostc and hostd) attached to the target CLARiiON array.
- The application runs 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.

These tests assume the following environment:



Testing 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 that is attached to the same array. In the Service Groups tab of the Cluster Explorer configuration tree, right-click the service group.
2. Click **Switch To**, and click the system that is attached to the same array (hostb) from the menu.
   The service group comes online on hostb and local volumes remain in the MIRRORED state.

3 Migrate the service group to a host that is attached to a different array. In the Service Groups tab of the Cluster Explorer configuration tree, right-click the service group.

4 Click **Switch To**, and click the system that is 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.

5 Accumulate dirty tracks on the new source-side and update them back on the target:
`C:\> hares -action mirrorview_res_name resync -sys hostc`
The variable *mirrorview_res_name* represents the name of the mirrorview resource.

6 After the devices transition to a source SYNCHRONIZED state, migrate the service group back to its original host. In the Service Groups tab of the Cluster Explorer configuration tree, right-click the service group.

7 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.

Testing 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 Shut down the host on which the application is running (hosta).
The service group fails over to hostb and devices are in the RW/SYNCHRONIZED state.

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

5 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.

Performing a disaster test

Test how robust your cluster is in case of a disaster.

To perform a disaster test

1 Shut down all hosts on the source side and shut down the source array.
If you can not shut down the source array, 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.

2 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.

3 In a global cluster, the administrator is notified of the failure. The administrator can then initiate the failover.

Performing the failback test

You can set up your cluster for a failback test.

To perform the failback test for asynchronous mode with consistency groups

1 Remove all the mirrors from 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, data corruption could occur if one of the LUNs took hardware damage and failed.

During this window, 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. This situation 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.

Failure scenarios

Review the failure scenarios and agent behavior in response to failure.

All host or all application failure

Even if both arrays are operational, the service group fails over in the following conditions:

- All hosts on the primary site side are disabled.
- The application cannot start successfully on any primary host.

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.

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

Before the MirrorView agent takes any action, it waits for the synchronization to complete in the following situations:

- The two arrays are healthy and the link that failed is restored.
- A failover is initiated while synchronization is in progress.

After the synchronization completes, the MirrorView agent runs 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.

Index

S

T

V