Cluster Server Agent for Network Appliance SnapMirror Installation and Configuration Guide

AIX, HP-UX, Linux, Solaris

7.0

VERITAS™

Cluster Server Agent for Network Appliance SnapMirror Installation and Configuration Guide

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

Agent Version: 7.0

Document version: 7.0 Rev 0

Legal Notice

Technical Support

Technical Support maintains support centers globally. Technical Support's primary role is to respond to specific queries about product features and functionality. The Technical Support group also creates content for our online Knowledge Base. The Technical Support group works collaboratively with the other functional areas within the company to answer your questions in a timely fashion.

Our support offerings include the following:

■ A range of support options that give you the flexibility to select the right amount of service for any size organization

■ Telephone and/or Web-based support that provides rapid response and up-to-the-minute information

■ Upgrade assurance that delivers software upgrades

■ Global support purchased on a regional business hours or 24 hours a day, 7 days a week basis

■ Premium service offerings that include Account Management Services

For information about our support offerings, you can visit our website at the following URL:

www.veritas.com/support

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

Contacting Technical Support

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

www.veritas.com/support

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

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

■ Product release level

■ Hardware information

■ Available memory, disk space, and NIC information

- Operating system

- Version and patch level

- Network topology

- Router, gateway, and IP address information

- Problem description:

  - Error messages and log files

  - Troubleshooting that was performed before contacting Technical Support

  - Recent software configuration changes and network changes

Licensing and registration

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

www.veritas.com/support

Customer service

Customer service information is available at the following URL:

www.veritas.com/support

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

- Questions regarding product licensing or serialization

- Product registration updates, such as address or name changes

- General product information (features, language availability, local dealers)

- Latest information about product updates and upgrades

- Information about upgrade assurance and support contracts

- Advice about technical support options

- Nontechnical presales questions

- Issues that are related to CD-ROMs, DVDs, or manuals

Support agreement resources

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

Worldwide (except Japan)	CustomerCare@veritas.com
Japan	CustomerCare_Japan@veritas.com

Contents

Introducing the agent for Network Appliance SnapMirror

This chapter includes the following topics:

■ About the agent for Network Appliance SnapMirror

■ Supported software

■ Supported hardware for Network Appliance SnapMirror

■ Typical Network Appliance SnapMirror setup in a VCS cluster

■ Network Appliance SnapMirror agent functions

About the agent for Network Appliance SnapMirror

The Cluster Server (VCS) agent for Network Appliance (NetApp) SnapMirror provides application failover support and recovery in environments that use SnapMirror to replicate data between NetApp filers.

The agent monitors and manages the state of a volume or a qtree for which SnapMirror replication is configured on the filers. The agent ensures that at any given time, only one host has safe exclusive access to the configured filer volume or qtree. It determines the current role of the filer volume or qtree with respect to replication, and promotes a read-only target to a read-write source during a wide-area failover. You can use the agent in global clusters that run VCS.

Note: The VCS agent for SnapMirror supports only the Network File Systems (NFS) protocol.

The agent supports SnapMirror with Data ONTAP (7-mode) as well as Clustered Data ONTAP (cluster-mode). However, it supports the following configurations only with 7-mode and not with cluster-mode:

- Qtree replication
- Multi-pathing
- Fire drill

Note: Cluster-mode support for this agent is available only on Linux. Also, only data protection relationships are supported with cluster-mode. Other relationships like vault and load-sharing are not supported.

See the following Technical Support TechNote for the latest updates or software issues for this agent:

http://www.veritas.com/docs/000030982

Supported software

For information on the software versions that the agent for Network Appliance SnapMirror supports, see the Veritas Services and Operations Readiness Tools (SORT) site: https://sort.veritas.com/agents.

Supported hardware for Network Appliance SnapMirror

The agent supports the following versions of Network Appliance hardware:

- Network Appliance FAS devices (any model)
- Network Appliance ONTAP 6.4 and later.

The agent does not support NetApp F800 filer series.

Typical Network Appliance SnapMirror setup in a VCS cluster

Figure 1-1 displays a typical cluster setup in a SnapMirror environment.

Figure 1-1   Typical clustering setup for the agent
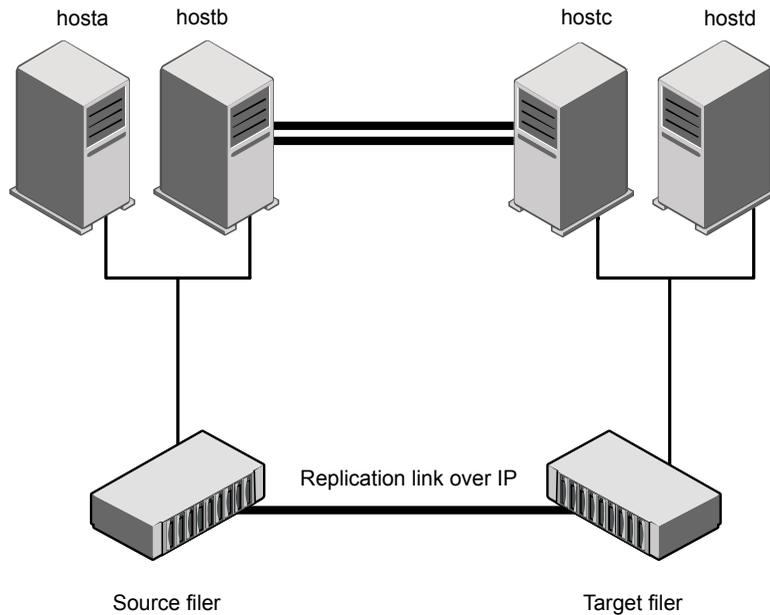
Clustering in a SnapMirror environment typically consists of the following hardware infrastructure:

- One or more source hosts are connected by an ethernet cable to a Network Appliance device. The Network Appliance SnapMirror agent supports NFS as a transport protocol.

- One or more target hosts that are attached to a second filer with similar characteristics as the source hosts. The volume or qtree are SnapMirror targets.

- Network heartbeating between the two data centers to determine their health; this network heartbeating could be LLT or TCP/IP.
  See "About cluster heartbeats" on page 27.

- In a global cluster environment, you must attach all hosts in the same cluster to the same filer.

Network Appliance SnapMirror agent functions

The Network Appliance SnapMirror agent monitors and manages the state of replicated Network Appliance volume or qtree that are attached to VCS nodes.

Agent functions bring resources online, take them offline, and perform different monitoring actions.

Table 1-1   Agent functions

Agent function	Description
online	Creates a lock file on the local host to indicate that the resource is online.
	Depending on whether the filer is source or target and the state of the specified volume or qtree on the filer, the agent performs the online function.
	See "About the Network Appliance SnapMirror agent's online function" on page 14.
offline	The agent removes the lock file. The agent does not perform any filer operations because taking the resource offline is not an indication of a pending role swap.
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 system on which this entry point is called. The agent prevents potential concurrency violation if the group fails over to another node.
	Note that the agent does not remove the lock file if the agent was started after a `hastop -force` command.
clean	Performs the same operation as offline.
action/PreSwitch	Ensures that the remote site cluster can come online during a planned failover within a GCO configuration. The VCS engine on the remote cluster invokes the PreSwitch action on all the resources of the remote site during a planned failover using the `hagrp -switch` command.
	For this, the PreSwitch attribute must be set to 1. The option `-nopre` indicates that the VCS engine must switch the servicegroup regardless of the value of the PreSwitch service group attribute.
	If running the PreSwitch action fails, the failover should not occur. This minimizes the application downtime and data loss.

Table 1-1   Agent functions *(continued)*

Agent function	Description
action/reestablish	Re-establishes the replication with a destination volume or qtree that is broken off. Invokes a 'snapmirror resync' to resynchronize contents with the source without doing a full transfer. This action also write-disables the destination volume or qtree.
actions/swap	Reverses the replication roles to make the original source the new destination and vice versa. It is intended for use when the original source comes back alive after an outage at the source location and the original destination is broken-off in the meantime to takeover the applications at the destination location. As part of this action, the new destination (i.e. the original source) is made to sync up with the new source.
actions/update	Used to run unscheduled updates. SnapMirror updates the destination automatically based on the update schedule specified in the /etc/snapmirror.conf. However, to run unscheduled updates, this update action may be invoked on the NetAppSnapMirror resource.
action/GetCurrentRPO	Fetches the current point-in-time RPO from the NetApp array in terms of SnapMirror lag. Invoke this action function on the DR system. **Note:** The agent does not store the computed RPO; make a note of the RPO for future reference.

The following functions are not supported in cluster-mode:

- action/PreSwitch

Note: In case of SnapMirror with cluster-mode, the agent bypasses the actions of this function and triggers the failover.

- action/reestablish
- actions/swap
- actions/update
- action/GetCurrentRPO

About the Network Appliance SnapMirror agent's online function

The agent's online entry point checks the state of the specified volume or qtree on the local filer device. The volume or qtree on the local filer is in the SOURCE state when it is the replication source. The volume or qtree is in the SNAPMIRRORED state when it is the replication target.

Note: In case of SnapMirror with cluster-mode, if a volume on the local filer is of the read-write type, the agent considers it to be the source for replication.

The agent performs the following online operations:

- If the state of the volume or qtree on the local filer device is source, the agent creates a lock file to indicate that the resource can come online

- If the filer is a target, the agent attempts to reverse the direction of replication. To reverse the replication, the agent promotes the local filer to a source and the original source to a target by performing the following operations:

  - Updates the destination volume or qtree

  - Quiesces' the replication

  - Breaks off the SnapMirror connection

  - Removes the SnapMirror schedule from the target filer and creates a new SnapMirror schedule on local filer

  - Resynchronizes the local filer with the new target filer

- If the original source is down, the agent attempts a takeover to enable local write access by performing the following operations:

  - Quiesces' the replication

  - Breaks off the SnapMirror connection

- If the original source comes online, you must resynchronize data manually

- If read-write access is enabled successfully, the online entry point touches a lock file

Installing and removing the agent for Network Appliance SnapMirror

This chapter includes the following topics:

- Before you install the agent for Network Appliance SnapMirror

- Installing the agent for Network Appliance SnapMirror

- Upgrading the agent for Network Appliance SnapMirror

- Removing the agent for Network Appliance SnapMirror

Before you install the agent for Network Appliance SnapMirror

Before you install the VCS agent for Network Appliance SnapMirror, ensure that you install and configure VCS on all nodes in the cluster.

Set up replication and the required hardware infrastructure. For information about setting up Oracle RAC environment, refer to the *Storage Foundation for Oracle RAC Configuration and Upgrade Guide*.

See "Typical Network Appliance SnapMirror setup in a VCS cluster" on page 10.

NetApp agent installation issue (dependency on the perl(XML::Parser) module)

The NetApp SnapMirror agent depends on the perl(XML::Parser) module. This dependency is inherited from NaServer, which is an SDK interface for SnapMirror. If the perl(XML::Parser) module is not installed on your operating system, then the agent installation will fail. Make sure that you install this module before installing the SnapMirror agent. For example, on the RHEL platform, the install will fail with the following error:

```
# rpm -ivh VRTSvcsna-7.0.0.0-Linux_GENERIC.noarch.rpm
```

Error for failed dependencies:

```
perl(XML::Parser) is needed by VRTSvcsna-7.0.0.0-Linux_GENERIC.noarch
```

Installing the agent for Network Appliance SnapMirror

You must install the Network Appliance SnapMirror 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 VCS.

To install the agent in a VCS environment

1   Download the Agent Pack from the Veritas Services and Operations Readiness Tools (SORT) site: https://sort.veritas.com/agents.

    You can download the complete Agent Pack tar file or the individual agent tar file.

2   Uncompress the file to a temporary location, say /tmp.

3   If you downloaded the complete Agent Pack tar file, navigate to the directory containing the package for the platform running in your environment.

AIX `cd1/aix/vcs/replication/ntap_agent/` *`agent_version`*`/pkgs/`

HP-UX `cd1/hpux/generic/vcs/replication/ntap_agent/`
(PA) *`agent_version`*`/PA/depot/`

HP-UX `cd1/hpux/generic/vcs/replication/ntap_agent/`
(IA) *`agent_version`*`/IA/depot`

Linux `cd1/linux/generic/vcs/replication/ntap_agent/`
`agent_version``/rpms/`

Solaris `cd1/solaris/dist_arch/vcs/replication/ntap_agent/`
`agent_version``/pkgs/`

If you downloaded the individual agent tar file, navigate to the pkgs directory (for AIX, HP-UX, and Solaris), or the rpms directory (for Linux).

4   Log in as a superuser.

5   Install the package.

AIX `# installp -ac -d VRTSvcsna.rte.bff VRTSvcsna.rte`

HP-UX `# swinstall -s 'pwd' VRTSvcsna`
(IA/PA)

Linux `# rpm -ihv \`
`VRTSvcsna-`*`AgentVersion`*`-Linux_GENERIC.noarch.rpm`

Solaris `# pkgadd -d . VRTSvcsna`

Upgrading the agent for Network Appliance SnapMirror

You must upgrade the agent on each node in the cluster.

To upgrade the agent software

1   Save the VCS configuration and stop the VCS engine.

```
# haconf -dump -makero
# hastop -all -force
```

2   Remove the agent from the node.

See "Removing the agent for Network Appliance SnapMirror" on page 18.

3   Delete the file /etc/VRTSvcs/conf/config/NetAppSnapMirrorTypes.cf.

4   Install the current version of the agent.

See "Installing the agent for Network Appliance SnapMirror" on page 16.

5   Copy the file NetAppSnapMirrorTypes.cf from the directory /etc/VRTSvcs/conf/ to the /etc/VRTSvcs/conf/config directory.

6   Repeat step 2 through step 5 on each node.

7   From a node in the cluster, edit your configuration file /etc/VRTSvcs/conf/config/main.cf.

Configure the new attributes, if applicable.

8   Verify the configuration by running the following command:

```
# hacf -verify config
```

9   Start VCS on local node first.

10   Start VCS on other nodes.

Removing the agent for Network Appliance SnapMirror

Before you attempt to remove the agent, make sure the application service group is not online.

You must remove the SnapMirror agent from each node in the cluster.

To remove the agent, type the following command on each node. Answer prompts accordingly:

AIX   # installp -u VRTSvcsna.rte

HP-UX   # swremove VRTSvcsna

Linux	`# rpm -e VRTSvcsna`
Solaris	`# pkgrm VRTSvcsna`

Note: To uninstall the agent IPS package on a Solaris 11 system:

```
# pkg uninstall VRTSvcsna
```

Configuring the agent for Network Appliance SnapMirror

This chapter includes the following topics:

Configuration concepts for Network Appliance SnapMirror agent

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

Resource type definition for the NetAppSnapMirror agent

The resource type definition defines the agent in VCS.

```
type NetAppSnapMirror (
        static int ActionTimeout = 120
        static keylist SupportedActions = { update,
        reestablish, swap, PreSwitch, GetCurrentRPO }
        static int MonitorInterval = 300
        static int OpenTimeout = 180
```

```
static int NumThreads = 1
static str ArgList[] = { VolumeName, SnapMirrorSchedule,
LocalFilerName, LocalFilerUserName, LocalFilerPword,
RemoteFilerName, RemoteFilerUserName, RemoteFilerPword,
OpsThrottle, VisibilityFrequency, VServer, CModeCluster }
str VolumeName
str SnapMirrorSchedule
str LocalFilerName
str LocalFilerUserName="root"
str LocalFilerPword
str RemoteFilerName
str RemoteFilerUserName="root"
str RemoteFilerPword
str OpsThrottle
int VisibilityFrequency = 180
int ComputeDRSLA = 1
str VServer
str CModeCluster
)
```

Attribute definitions for the NetAppSnapMirror agent

Review the description of the agent attributes. You must assign values to the following attributes.

Required attributes

Note: Some required attributes are applicable only to configurations with Clustered Data ONTAP.

Table 3-1   Required attributes for the SnapMirror agent

Attribute	Description
Name: LocalFilerName	The DNS resolvable name of the local NetApp filer.
Type: string Dimension: scalar	You must use the same value for the LocalFilerName as in the NetAppSnapMirror configuration. If the filer's name is not fully qualified in the NetAppSnapMirror configuration, do not use the fully qualified name to set this attribute.
	For example, if the filer name is defined as netapp3.example.com in the configuration, use the same filer's name to set the attribute.

Table 3-1   Required attributes for the SnapMirror agent *(continued)*

Attribute	Description
Name: LocalFilerUserName Type: string Dimension: scalar	The local NetApp filer user who has permissions to perform SnapMirror actions using ONTAP SDK.
Name: LocalFilerPword Type: string Dimension: scalar	A valid encrypted password for the user on the local filer. If you enter the password from the command line, run the `vcsencrypt` command to encrypt the password. Do not encrypt passwords when entering them from the GUI. **Note:** In a 7-mode configuration, if you do not specify a password, the agent uses the hosts.equiv file on the filer to determine the access rights.
Name: RemoteFilerName Type: string Dimension: scalar	The DNS resolvable name of the remote NetApp filer. You must use the same value for the RemoteFilerName as in the NetAppSnapMirror configuration. If the filer's name is not fully qualified in the NetAppSnapMirror configuration, do not use the fully qualified name to set this attribute. For example, if the filer name is defined as netapp3.example.com in the configuration, use the same filer's name to set the attribute.
Name: RemoteFilerUserName Type: string Dimension: scalar	Remote NetApp filer user name: Have a permission to perform SnapMirror actions using ONTAP SDK.
Name: RemoteFilerPword Type: string Dimension: scalar	A valid encrypted password for the root user on the remote filer. If you enter the password from the command line, run the `vcsencrypt` command to encrypt the password. Do not encrypt passwords when entering them from the GUI. **Note:** In a 7-mode configuration, if you do not specify a password, the agent uses the hosts.equiv file on the filer to determine the access rights.

Table 3-1   Required attributes for the SnapMirror agent *(continued)*

Attribute	Description
Name: VolumeName Type: string Dimension: scalar	Specifies the name of the replicated volume or qtree on the filer. You must use same case for the volume name as on the filer because the attribute value is case-sensitive. **Note:** For qtrees, you must use the full path name, for example, `/vol/myvol/myqtree`.
Name: VServer Type: string Dimension: scalar	(Cluster-mode only) Specifies the Vserver name of the SnapMirror relationship in cluster-mode.
Name: CModeCluster Type: string Dimension: scalar	(Cluster-mode only) Specifies the cluster name of the SnapMirror relationship in cluster-mode.

Optional attributes

Note: The optional attributes are not applicable to cluster-mode configurations.

Table 3-2   Optional attributes for the SnapMirror agent:

Attribute	Description
Name: OpsThrottle Type: string Dimension: scalar	Specifies how out of sync the local and remote filer can run. You must set this attribute to configure SnapMirror in synchronous mode. You can specify the attribute value in one of the following units: ■ ops - Number of I/O operations ■ s - Number of seconds ■ ms - Number of milliseconds Example: "5ops" Note that to configure SnapMirror in synchronous mode, you must set the value as "0s", "0ms" or "0ops".

Table 3-2   Optional attributes for the SnapMirror agent: *(continued)*

Attribute	Description
Name: SnapMirrorSchedule Type: string Dimension: scalar	Specifies the schedule that the destination uses for updating data. This attribute can be left empty to configure SnapMirror in synchronous mode. You must set this attribute to configure SnapMirror in asynchronous mode. Example: 0-59/5 * * *
Name: VisibilityFrequency Type: integer Dimension: scalar	Specifies the time when the source snapshot is visible at the destination mirror. Default: 180 seconds
Name: ComputeDRSLA Type: integer Dimension: scalar	Used to enable Recovery Point Objective (RPO) computation. This attribute cannot be edited. Default value : 1

Sample configuration for the Network Appliance SnapMirror agent

You can configure the SnapMirror agent with the Mount agent. You can also use the NetAppFiler and NetAppExport agents provided by Network Appliance.

Sample configuration for the Network Appliance SnapMirror agent

- Sample configuration with the Mount agent

```
Mount ntap (
    MountPoint = "/ntap"
    BlockDevice = "netapp1:/vol/unixvol"
    FSType = nfs
    FsckOpt = "-n"
    )

    NetAppSnapMirror sm (
    VolumeName = unixvol
    SnapMirrorSchedule = "0-59/5 * * *"
    LocalFilerName = netapp1
    LocalFilerUserName = root
    LocalFilerPword = xxxx
    RemoteFilerName = netapp2
    RemoteFilerUserName = root
```

```
        RemoteFilerPword = xxxx
        )

    ntap requires sm
```

- Sample configuration with the NetAppExport and NetAppFiler agents

```
group ntap (
        SystemList = { primary1 = 0, primary2 = 1 }
        ClusterList = { secondary = 0, primary = 1 }
        )

        NetAppExport nfs (
            FilerResName = filer
            FilerPathName = "/vol/unixvol"
            FilerOptions @primary1 = "access=primary1,root=primary1"
            FilerOptions @primary2 = "access=primary2,root=primary2"
            MountPathName = "/ntap"
            MountOptions = rw
            )
        NetAppFiler filer (
            FilerName = priflr
            )

        NetAppSnapMirror sm (
            VolumeName = unixvol
            SnapMirrorSchedule = "0 23 * *"
            LocalFilerName = priflr
            LocalFilerUserName = root
            LocalFilerPword = xxxx
            RemoteFilerName = secflr
            RemoteFilerUserName = root
            RemoteFilerPword = xxxx
            )

        nfs requires sm
        sm requires filer
```

Sample configuration for the Network Appliance SnapMirror agent in cluster-mode:

```
group SG_NTAP (
    SystemList = { dl380g9-50vm5 = 0, dl380g9-50vm6 = 1 }
    ClusterList = { PRIM_CLUS = 0, SEC_CLUS = 1 }
    Authority = 1
```

```
AutoStartList = { dl380g9-50vm5 }
ClusterFailOverPolicy = Auto
)

Mount res_mnt (
    MountPoint = "/mnt1"
    BlockDevice =
    "10.209.92.147:/NAS_IAFAS3240_vol_n2_sr226876_DL380G9_50vm5_8_2gb_1_m
    FSType = nfs
    FsckOpt = "-n"
    )

NetAppSnapMirror res_ntp (
    VolumeName =
    NAS_IAFAS3240_vol_n2_sr226876_DL380G9_50vm5_8_2gb_1_mirror
    LocalFilerName = "10.209.92.171"
    LocalFilerUserName = vrpadmin
    LocalFilerPword = xxxx
    RemoteFilerName = "10.209.92.170"
    RemoteFilerUserName = vrpadmin
    RemoteFilerPword = xxxx
    CModeCluster = IA2552-02
    VServer = NAS_IAFAS2552
    )

res_mnt requires res_ntp

// resource dependency tree
//
//   group SG_NTAP
//   {
//   Mount res_mnt
//     {
//         NetAppSnapMirror res_ntp
//     }
//   }
```

Before you configure the agent for Network Appliance SnapMirror

Before you configure the agent, review the following information:

- Verify that you have installed the agent on all systems in the cluster.
- Verify the hardware setup for the agent.
  See "Typical Network Appliance SnapMirror setup in a VCS cluster" on page 10.
- Make sure that the cluster has an effective heartbeat mechanism in place.
  See "About cluster heartbeats" on page 27.

You can automate the mounting of filer volumes or qtrees through VCS. Configure a resource of type Mount as the parent of the NetAppSnapMirror resource and set the FSType attribute of the Mount resource to nfs. You must also configure the filer to export the volumes or qtrees to all the nodes in the cluster.

About cluster heartbeats

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.

Configuring the agent for Network Appliance SnapMirror

You can adapt most clustered applications to a disaster recovery environment.

If you plan to configure the agent in a global cluster, make sure that you have configured the global service group for the application.

After configuration, the application service group must follow the dependency diagram.

See "Sample configuration for the Network Appliance SnapMirror agent" on page 24.

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 (Java Console) and log on to the cluster.

2 If the agent resource type (NetAppSnapMirror) is not added to your configuration, add it. From the Cluster Explorer **File** menu, choose **Import Types**, and select:

/etc/VRTSvcs/conf/NetAppSnapMirrorTypes.cf

3 Click **Import**.

4   Save the configuration.

5   Add a resource of type NetAppSnapMirror at the bottom of the service group.

6   Configure the attributes of the NetAppSnapMirror resource.

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

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

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

9   Repeat step 5 through step 8 for each service group in each cluster that uses replicated data.

Configuring the agent to compute RPO

In a global cluster environment, the agent for Network Appliance SnapMirror can compute the recovery point objective (RPO), which is a disaster recovery (DR) SLA. In a DR configuration where data is replicated asynchronously to the DR site, the DR site data is not always as current as the primary site data.

RPO is the maximum acceptable amount of data loss in case of a disaster at the primary site. The agent computes RPO in terms of time, that is, in seconds.

Before you configure the agent to compute the RPO, ensure that the following pre-requisites are met:

■   The service group containing the NetAppSnapMirror resource and the VxVM disk group resource are online at the production site.

■   The disk group resource is dependent on the NetAppSnapMirror resource.

To configure the agent to compute the RPO:

1   In the DR cluster, on any one of the nodes where devices are asynchronously replicated and where the service group is configured, run the following command to start the RPO computation:

```
hares -modify NetAppSnapMirror_resource_name ComputeDRSLA 1 -sys
system_name.
```

2   Run the following command on the same node in the DR cluster:

```
hares -action NetAppSnapMirror_resource_name GetCurrentRPO -sys
system_name
```

The action entry point displays the RPO. The agent does not store the computed RPO; make a note of the RPO for future reference.

If the RPO is not reported, it indicates that the agent needs more time to finish computing the RPO. Wait for some more time before you run the GetCurrentRPO action function again.

3   To stop RPO computation, run the following command:

```
hares -modify NetAppSnapMirror_resource_name ComputeDRSLA 0 -sys
system_name
```

Configuring the agent for SnapMirror replication over multiple paths

The agent for Network Appliance SnapMirror supports multiple paths for replication between source and destination filer. The agent can work for up to two paths for a given SnapMirror relationship. Agent detects configured multiple paths on secondary filer and can handle it accordingly during a failover. The agent supports both Multiplexing mode and Failover mode of SnapMirror over multiple paths. The paths can be Ethernet, Fibre Channel or a combination of Ethernet and Fibre Channel.

We recommend ONTAP 7.3.3 for agent support for SnapMirror over Multiple paths. ONTAP versions before version 7.3.3 needs patch from NetApp to fix issue in ONTAP API `snapmirror-set-connection`.

Configuring multiple paths in SnapMirror

Set up SnapMirror over multiple paths on destination filer. Ensure valid paths between source filer and destination filer. Edit the `snapmirror.conf` file on the destination system to add a connection line and relation entry to specify the volumes or qtrees to be copied and the schedule on which the destination is updated. Use

the connection name specified as the source system. The format of the line in snapmirror.conf as follows:

```
name = failover (src_system1, dest_system1) (src_system2,dest_system2)
#connection entry name:srcvol1 dest_filer:destvol1 - 0-55/5 * * *
#Relationship entry
```

The agent behavior when SnapMirror over multiple path is configured

Filer1 is source having two interfaces:`source_ip_addr1` and `source_ip_addr2`. Filer2 is a destination filer having interfaces: `dest_ip_addr1` and `dest_ip_addr2`. Two SnapMirror relations are configured between srcvol1 to destvol1 and srcvol2 to destvol2 using failover mode of multiple paths. Following table describes agent behavior:

Table 3-3   Agent behaviour

Action	Source system	Destination system
Initial setup: There are two actively running SnapMirror relations between filer1 and filer2 with reference to connection My-conn. Agent has relations configured as resource1 and resource2 accordingly.		`My-conn = failover` `(source_ip_addr1,dest_ip_addr1)` `(source_ip_addr2,dest_ip_addr2)` `My-conn:srcvol1` `filer2:destvol1 - 0-55/5` `* * *` `My-conn:srcvol2` `filer2:destvol2 - 0-55/5` `* * *`

Table 3-3   Agent behaviour *(continued)*

Action	Source system	Destination system

First relation(resource1) is migrated to Filer2: Agent detects multiple paths configured and derived connection name with resource1. Agent adds new connection entry My-conn to filer1 if existing connection is not there with same name. If there is existing connection with name My-conn, it will cross check if they are identical.	`My-conn = failover`
	`(dest_ip_addr1,`
	`source_ip_addr1)`
	`(dest_ip_addr2,`
	`source_ip_addr2)`
	
	`My-conn:destvol1`
	`filer1:sourcevol1 -`
	`0-55/5 * * *`
		`My-conn = failover`
		`(source_ip_addr1,dest_ip_addr1)`
		`(source_ip_addr2,dest_ip_addr2)`
		
		`My-conn:srcvol2`
		`filer2:destvol2 - 0-55/5`
		`* * *`

First relation is migrated back to filer1: Check and add relationship entry. The agent does not delete connection entry on filer1.	`My-conn = failover`
	`(dest_ip_addr1,`
	`source_ip_addr1)`
	`(dest_ip_addr2,`
	`source_ip_addr2)`
		`My-conn = failover`
		`(source_ip_addr1,dest_ip_addr1)`
		`(source_ip_addr2,dest_ip_addr2)`
		
		`My-conn:srcvol2`
		`filer2:destvol2 - 0-55/5`
		`* * *`
		
		`My-conn:srcvol1`
		`filer2:destvol1 - 0-55/5`
		`* * *`

Migrating both relations to Filer2: Check and add relationship entry. The agent won't delete connection entry on filer2.	`My-conn = failover`
	`(dest_ip_addr1,`
	`source_ip_addr1)`
	`(dest_ip_addr2,`
	`source_ip_addr2)`
	
	`My-conn:destvol1`
	`filer1:sourcevol1 -`
	`0-55/5 * * *`
	
	`My-conn:destvol2`
	`filer1:sourcevol2 -`
	`0-55/5 * * *`
		`My-conn = failover`
		`(source_ip_addr1,dest_ip_addr1)`
		`(source_ip_addr2,dest_ip_addr2)`

Managing and testing clustering support for Network Appliance SnapMirror

This chapter includes the following topics:

- Performing a disaster test
- Failure scenarios for Network Appliance SnapMirror

Performing a disaster test

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

To perform a disaster test

1   Shut down the host on which the service group is online as well as the filer to which it is attached.

   If you cannot shut down the filer, disconnect it from the network to make it invisible to the remote site.

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

3   Reboot the host and restart the filer. After the filer comes up, issue the swap action to resynchronize all data back to the source. After the resynchronization completes, switch the application back to the primary node.

Failure scenarios for Network Appliance SnapMirror

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

Site disaster

In a global cluster environment, if the site running the application fails, VCS detects the outage and recommends failing over to the DR site.

The agent for SnapMirror attempts a data migration, which means taking over the target data and reversing the role of replication. The data migration during a site disaster involves:

- Breaking the replication to enable write access to the target (through the NetAppSnapMirror agent).

- Updating the SnapMirror configuration to demote the original source to a target, and begin replicating from the new source (also through the NetAppSnapMirror agent).

- Starting the application services through the VCS agents for various applications.

If the original filer returns, then you must resynchronize the filer.

All host or all application failure

In global cluster environments, multiple service groups can fail over in parallel.

In a global cluster environment, VCS detects the outage and recommends failing over to the DR site in one of the following cases:

- The primary host that runs the application fails

- The application itself fails

In this case, the VCS agents for SnapMirror attempt a data migration because both filers are healthy. The agent takes over the target data and reverses the role of replication. The data migration during a host or application failure involves:

- Breaking the replication to enable write access to the target (through the NetAppSnapMirror agent).

- Updating the SnapMirror configuration to demote the original source to a target, and begin replicating from the new source (also through the NetAppSnapMirror agent).

- Starting the application services through the VCS agents for various applications.

Setting up a fire drill

This chapter includes the following topics:

- About fire drills

- About the NetAppSnapMirrorSnap agent

- Before you configure the fire drill service group

- Configuring the fire drill service group

- Verifying a successful fire drill

- Sample configuration for a fire drill service group

About fire drills

A fire drill procedure verifies the fault-readiness of a disaster recovery configuration. This procedure is done without stopping the application at the primary site and disrupting user access.

A fire drill is performed at the secondary site using a special service group for fire drills. The fire drill service group is identical to the application service group, but uses a fire drill resource in place of the replication agent resource. The fire drill service group uses a copy of the data that is used by the application service group.

In clusters employing Network Appliance SnapMirror, the NetAppSnapMirrorSnap resource manages the replication relationship during a fire drill.

Bringing the fire drill service group online demonstrates the ability of the application service group to come online at the remote site when a failover occurs.

Note: The NetAppSnapMirrorSnap agent does not support fire drills for qtree.

About the NetAppSnapMirrorSnap agent

The NetAppSnapMirrorSnap agent is the fire drill agent for Network Appliance SnapMirror.

The agent manages the replication relationship between the source and target arrays when running a fire drill. Configure the NetAppSnapMirrorSnap resource in the fire drill service group, in place of the NetAppSnapMirror resource.

The agent is configured in the fire drill service group and supports fire drills only for flexvols. The agent exports configured volumes as nfs shares.

NetAppSnapMirrorSnap agent functions

The NetAppSnapMirrorSnap agent performs the following functions:

Table 5-1   Agent functions

Function	Description
online	■ Takes a clone of the target volume that is used by the application. ■ Breaks the mirror relation between the source and target volumes. ■ Takes a clone of the target volume and exports it to bring up the application. ■ Breaks the mirror relation between the source and target volumes. ■ Exports the target volume without taking a snapshot. ■ Creates a lock file to indicate that the resource is online. After the online operation is completed, the following nfs share becomes available: <Filename>:/vol/<volname>_fd
offline	Reestablishes the mirror relationship between the source and target volumes by doing a resync. Removes the lock file created by the Online operation.
monitor	Verifies the existence of the lock file to make sure the resource is online.
clean	■ Reestablishes the mirror relationship between the source and target volumes by doing a resync. ■ Removes the lock file created by the Online operation.

Table 5-1   Agent functions *(continued)*

Function	Description
open	If the lock file does not exist, the agent takes no action.
	If the lock file exists:
	■ If any resources that depend on the NetAppSnapMirrorSnap are online, the agent does not take any action.
	■ If any resources that depend on the NetAppSnapMirrorSnap are not online, the agent removes the lock file.

Resource type definition for the NetAppSnapMirrorSnap agent

Following is the resource type definition for the NetAppSnapMirrorSnap agent:

```
type NetAppSnapMirrorSnap (
    static int ActionTimeout = 120
    static keylist SupportedActions = { }
    static int MonitorInterval = 300
    static int NumThreads = 1
    static str ArgList[] = { TargetResName, FilerName,
    FilerPword, MountSO, UseSO, RequiredSO, SnapshotName}
    str TargetResName
    str FilerName
    str FilerPword
    str MountSO
    str UseSO
    str RequiredSO
    str SnapshotName
    )
```

Attribute definitions for the NetAppSnapMirrorSnap agent

To customize the behavior of the NetAppSnapMirrorSnap agent, configure the following attributes:

Table 5-2   Agent attributes

Attribute	Description
TargetResName	The name of the VCS resource that manages the object to be snapshot. This can be a SnapMirror type resource or a NetApp type resource.
	Type-Dimension: string-scalar
FilerName	The name of the filer to which the target volumes belong.
	Type-Dimension: string-scalar
FilerPword	The password to access the filer specified by the attribute FilerName.
	Type-Dimension: string-scalar
UseSnapshot	Specifies whether the NetAppSnapMirrorSnap resource takes a local snapshot of the target array. Set this attribute to 1.
	Type-Dimension: integer-scalar
RequireSnapshot	Specifies whether the NetAppSnapMirrorSnap resource must take a snapshot before coming online.
	Set this attribute to 1 if you want the resource to come online only after it succeeds in taking a snapshot.
	Type-Dimension: integer-scalar
	Note: Set this attribute to 1 only if UseSnapshot is set to 1.
MountSnapshot	Specifies whether the resource uses the snapshot to bring the service group online. Set this attribute to 1.
	Type-Dimension: integer-scalar
	Note: Set this attribute to 1 only if the UseSnapshot attribute is set to 1.
SnapshotName	For Gold configuration only.
	Specifies the name of the snapshot used to clone the target volume. For Example: hourly.0, nightly.0.
	Type-Dimension: string-scalar

Before you configure the fire drill service group

Before you configure the fire drill service group, ensure that the following pre-requisites are met:

- Make sure the application service group is configured with a NetAppSnapMirror resource.

- Make sure the infrastructure to take volume clones is properly configured between the source and target volumes.

- You must have a NetApp FlexClone license as well as NFS licensed and enabled to test the creation of a snapshot of the target flexvol.

Configuring the fire drill service group

On the secondary site, the initial steps create a fire drill service group that closely follows the configuration of the original application service group. The fire drill service group uses a point-in-time copy of the production data. Bringing the fire drill service group online on the secondary site demonstrates the ability of the application service group to fail over and come online at the secondary site, should the need arise.

See "Sample configuration for a fire drill service group" on page 41.

You can create the fire drill service group using one of the following methods:

- Cluster Manager (Java Console)
  See "Creating the fire drill service group using Cluster Manager (Java Console)" on page 38.

- Fire Drill Setup wizard
  This text-based wizard is available at /opt/VRTSvcs/bin/fdsetup-ntap.
  See "Creating the fire drill service group using the Fire Drill SetUp Wizard" on page 40.

Creating the fire drill service group using Cluster Manager (Java Console)

This section describes how to use Cluster Manager (Java Console) to create the fire drill service group. After creating the fire drill service group, you must set the failover attribute to false so that the fire drill service group does not fail over to another node during a test.

To create the fire drill service group

1 Open the Cluster Manager (Java Console).

2 Log on to the cluster and click **OK**.

3   Click the **Service Group** tab in the left pane and click the **Resources** tab in the right pane.

4   Right-click the cluster in the left pane and click **Add Service Group**.

5   In the **Add Service Group** dialog box, provide information about the new service group.

    ■   In Service Group name, enter a name for the fire drill service group.

    ■   Select systems from the Available Systems box and click the arrows to add them to the Systems for Service Group box.

    ■   Click **OK.**

To disable the AutoFailOver attribute

1   Click the **Service Group** tab in the left pane and select the fire drill service group.

2   Click the **Properties** tab in the right pane.

3   Click the **Show all attributes** button.

4   Double-click the **AutoFailOver** attribute.

5   In the **Edit Attribute** dialog box, clear the **AutoFailOver** check box.

6   Click **OK** to close the **Edit Attribute** dialog box.

7   Click the **Save and Close Configuration** icon in the toolbar.

Adding resources to the fire drill service group

Add resources to the new fire drill service group to recreate key aspects of the application service group.

To add resources to the service group

1   In Cluster Explorer, click the **Service Group** tab in the left pane, click the application service group and click the **Resources** tab in the right pane.

2   Right-click the resource at the top of the tree, select **Copy > Self and Child Nodes**.

3   In the left pane, click the fire drill service group.

4   Right-click the right pane, and click **Paste**.

5   In the **Name Clashes** dialog box, specify a way for the resource names to be modified, for example, insert an '_fd' suffix. Click **Apply**.

6   Click **OK**.

Configuring resources for fire drill service group

Edit the resources in the fire drill service group so they work properly with the duplicated data. The attributes must be modified to reflect the configuration at the remote site. Bringing the service group online without modifying resource attributes is likely to result in a cluster fault and interruption in service.

To configure the fire drill service group

1   In Cluster Explorer, click the **Service Group** tab in the left pane.

2   Click the fire drill service group in the left pane and click the **Resources** tab in the right pane.

3   Right-click the NetAppSnapMirror resource and click **Delete**.

4   Add a resource of type NetAppSnapMirrorSnap and configure its attributes.

5   Right-click the resource to be edited and click **View > Properties View**. If a resource to be edited does not appear in the pane, click **Show All Attributes**.

6   Edit attributes to reflect the configuration at the remote site. For example, change the Mount resources so that they point to the volumes that are used in the fire drill service group.

Creating the fire drill service group using the Fire Drill SetUp Wizard

This section describes how to use the Fire Drill SetUp Wizard to create the fire drill service group.

To create the fire drill service group

1   Start the Fire Drill SetUp Wizard.

    `/opt/VRTSvcs/bin/fdsetup-ntap`

2   Enter the name of the application service group for which you want to configure a fire drill service group.

3   Select the supported snapshot configurations:

    Gold, Silver, or Bronze

4   Choose whether to run a Bronze fire drill, if the snapshot fails with Gold or Silver configurations.

    `If snapshot fails, should bronze be used? [y,n,q]`**(n)**

5   Press **Return** to verify the snapshot infrastructure.

6   Enter **y** to create the fire drill service group.

    The wizard runs various commands to create the fire drill service group.

7 In Linux clusters, verify that the StartVolumes attribute for each DiskGroup type resource in the fire drill group is set to 1. If not, modify the resource to set the value to 1.

8 Schedule fire drill for the service group by adding the following command to the crontab to be run at regular intervals.

```
/opt/VRTSvcs/bin/fdsched-netappsnapmirror
```

9 Make fire drill highly available by adding the following command to the crontab on every node in this cluster.

```
fdsched-netappsnapmirror
```

Verifying a successful fire drill

Run the fire drill routine periodically to verify the application service group can fail over to the remote node.

To verify a successful fire drill

1 Bring the fire drill service group online on a node at the secondary site that does not have the application running.

If the fire drill service group comes online, it action validates your disaster recovery configuration. The production service group can fail over to the secondary site in the event of an actual failure (disaster) at the primary site.

2 If the fire drill service group does not come online, review the VCS engine log for more information.

3 Take the fire drill offline after its functioning has been validated.

Failing to take the fire drill offline could cause failures in your environment. For example, if the application service group fails over to the node hosting the fire drill service group, there would be resource conflicts, resulting in both service groups faulting.

Sample configuration for a fire drill service group

The sample configuration of a fire drill service group is identical to an application service group with a hardware replication resource. However, in a fire drill service group, the NetAppSnapMirrorSnap resource replaces the NetAppSnapMirror resource.

In the mount resource, the name of the volume gets replaced by `<volname>_fd`.

You can configure a resource of type NetAppSnapMirrorSnap in the main.cf file as follows:

```
group firedrill_sg (
SystemList = { thoribm90mp4 = 0 }
)
Mount firedrill_mount (
MountPoint = "/ntap_sec_fd"
BlockDevice = "netapp3:/vol/gmvolsec_aix_fd"
FSType = nfs
)
NetAppSnapMirrorSnap ntap_sec_fd (
TargetResName = ntap_sec
FilerName = netapp3
FilerPword = xxxx
MountSnapshot = 1
UseSnapshot = 0
RequireSnapshot = 0
)
firedrill_mount requires ntap_sec_fd

// resource dependency tree
//
// group firedrill_sg
// {
// Mount firedrill_mount
// {
// NetAppSnapMirrorSnap ntap_sec_fd
// }
// }
```

Index