

Veritas Cluster Server Agents for Veritas FlashSnap™ Agent for Symmetrix Installation And Configuration Guide

Solaris

5.0

Veritas Cluster Server Agents for Veritas FlashSnap Agent for Symmetrix Installation And Configuration Guide

Copyright © 2006 Symantec Corporation. All rights reserved.

Veritas FlashSnap™ Agent 5.0 for Symmetrix

Symantec, the Symantec logo, and FlashSnap are trademarks or registered trademarks of Symantec Corporation or its affiliates in the U.S. and other countries. Other names may be trademarks of their respective owners.

The product described in this document is distributed under licenses restricting its use, copying, distribution, and decompilation/reverse engineering. No part of this document may be reproduced in any form by any means without prior written authorization of Symantec Corporation and its licensors, if any.

THIS DOCUMENTATION IS PROVIDED “AS IS” AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID, SYMANTEC CORPORATION SHALL NOT BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING PERFORMANCE, OR USE OF THIS DOCUMENTATION. THE INFORMATION CONTAINED IN THIS DOCUMENTATION IS SUBJECT TO CHANGE WITHOUT NOTICE.

The Licensed Software and Documentation are deemed to be “commercial computer software” and “commercial computer software documentation” as defined in FAR Sections 12.212 and DFARS Section 227.7202.

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.

Solaris is a trademark of Sun Microsystems, Inc.

Licensing and registration

Veritas FlashSnap Agent for Symmetrix is a licensed product. See the Veritas FlashSnap Agent for Symmetrix Installation Guide for license installation instructions.

Technical support

For technical assistance, visit <http://support.veritas.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	Veritas FlashSnap Agent for Symmetrix HA Agents	
	What's in this manual	7
	Getting help	7
	Veritas FlashSnap Agent for Symmetrix HA agents	8
	List of VxFAS HA agents	9
	VCS Cluster components	9
	Resources	9
	Attributes	9
	Service groups	9
	Installing VxFAS HA agents	10
	Upgrading to VxFAS HA Agents 5.0	10
	Configuring VxFAS HA agents	10
	Modifying the agents and their resources	11
	Resources and Service groups for VxFAS HA feature	12
	Resource dependencies	13
	Activating VxFAS in the VCS environment	15
	Guidelines for creating service groups and resources	15
	Creating a service group	16
	Creating a resource instance for a VxSymDevGrp resource	17
	Creating a resource instance for a VxSymLog resource	18
	Creating a resource instance for a DiskGroup resource	19
	Creating a resource instance for a mount resource	20
	Creating a resource instance for a VxSymRecover resource	21
Chapter 2	Operational Notes	
	Symmetrix-specific setup	23
	Modifying the VxSymLog LogDiskList attribute	24
	VxFAS HA operational notes	26
	VxFAS command-specific issues	26
	Logging-specific issues	27
	VxVM-specific issues	29
	VCS-specific issues	29
	EMC PowerPath-specific issues	32
	Re-using devices from other device groups	33
	Converting non-HA setup to HA	35

Chapter 3

VxFAS Agent Definitions

VxSymDevGrp agent	38
Type definition	39
Sample configuration	39
VxSymLog agent	40
Type definition	41
Sample configuration	41
VxSymRecover agent	42
Type definition	42
Sample configuration	42
DiskGroup agent	43
Type definition	43
Sample configuration	43
Mount agent	44
Type definition	45
Sample configuration	45
Example main.cf file	46

Index

Veritas FlashSnap Agent for Symmetrix HA Agents

This guide describes how to install and configure the components of the Veritas FlashSnap Agent for Symmetrix (VxFAS) high availability (HA) agents in an EMC Symmetrix environment. This guide also explains how to create, modify, and activate Veritas Cluster Server (VCS) resources and service groups.

What's in this manual

- [Chapter 1. “Veritas FlashSnap Agent for Symmetrix HA Agents” on page 7](#) introduces the concepts of VCS resources, attributes, and service groups. Describes VxFAS HA agent installation and configuration. Provides guidelines for creating service groups and resources.
- [Chapter 2. “Operational Notes” on page 23](#) describes how to set up a VCS cluster using a Symmetrix disk array as shared storage and discusses issues related to VxFAS commands, VCS, and EMC PowerPath configuration and recovery.
- [Chapter 3. “VxFAS Agent Definitions” on page 37](#) provides the programmatic definitions of the VxFAS HA agents and the VCS bundled agents required for the high availability product to operate.

Getting help

For technical assistance, visit <http://support.veritas.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.

Diagnostic tools are also available to assist in troubleshooting problems associated with the product. These tools are available on disc or can be downloaded from the Veritas FTP site. See the `README.VRTSspt` file in the `/support` directory for details.

For information on purchasing product documentation, visit <http://webstore.veritas.com>.

Unique Message Number

If you encounter a product error message, record the unique message number preceding the text of the message. When contacting Symantec Technical Support, either by telephone or by visiting the Technical Support website, be sure to provide the relevant message number. Technical Support will use this message number to quickly determine if there are TechNotes or other information available for you.

A unique message number is an alpha-numeric string beginning with the letter “V”. For example, in the message number:

```
V-5-732-8018 At least one disk must be specified
```

the “V” indicates that this is a Veritas product error message. The text of the error message follows the unique message number.

Veritas FlashSnap Agent for Symmetrix HA agents

Agents are processes that manage predefined resource types. When an agent is started, it obtains configuration information from the Veritas Cluster Server (VCS). It then periodically monitors the resources and updates VCS with the resource status. Typically agents:

- Bring resources online
- Take resources offline
- Monitor resources and report any state changes to VCS

VCS bundled agents are part of VCS and are installed when VCS is installed (as part of `VRTSvcsag` package). The VxFAS HA agents are add-on resources to VCS for the Veritas FlashSnap™ Agent for Symmetrix. VxFAS HA agents and resource types are part of the `VRTStfag` package and are added when you install the package and run the `vxtfvcsconf` script.

For more information on VCS agents, see the *Veritas Cluster Server Bundled Agents Reference Guide*. PDF versions of this guide are located under the `/opt/VRTSvcsdc` directory after installing the `VRTSvcsdc` package.

List of VxFAS HA agents

The VxFAS HA agents include:

- [VxSymDevGrp agent](#)
- [VxSymLog agent](#)
- [VxSymRecover agent](#)

The HA feature requires these VCS bundled agents:

- [DiskGroup agent](#)
- [Mount agent](#)

VCS Cluster components

Resources, attributes, and service groups are components integral to cluster functionality (see the *Veritas Cluster Server User's Guide* in the `/opt/VRTSvcsdc` directory for more information).

Resources

Resources are hardware or software entities, such as disks, volumes, file system mount points, network interface cards (NICs), IP addresses, applications, and databases. Resources work together to provide a service to clients in a client/server environment. Resource types are defined in the `types.cf` file by a collection of attributes. The VCS configuration file, `main.cf`, contains the values for the attributes of the resources. The `main.cf` file incorporates the resources listed in the `types.cf` by way of an `include` directive.

Attributes

Attributes contain data regarding the cluster, systems, service groups, resources, resource types, and agents. Each attribute has a definition and a value. A user defines an attribute by specifying its data type and dimension. A specified value for a given attribute configures the resource to function in a specific way. By modifying the value of an attribute of a resource, a user can change the way the VCS agent manages that resource. Attributes also have default values that are assigned when a value is not specified.

Service groups

Service groups are comprised of related resources. When a service group is brought online, all the resources within the group are brought online.

Installing VxFAS HA agents

The `VRTSfasag` (Veritas Cluster Server Agents for VxFAS) package includes the VxFAS HA agents (see “[List of VxFAS HA agents](#)” on page 9). Installation of the `VRTStfag` package requires the following packages:

- `VRTSvcs` (Veritas Cluster Server) version 4.1
- `VRTSvcsag` (Veritas Cluster Server Bundled Agents) version 4.1
- `VRTSfas` (Veritas FlashSnap Agent for Symmetrix) version 4.1

Install these packages first and then install the `VRTSfasag` package using the `pkgadd` command.

`VRTSvcs` and `VRTSfas` may have their own dependencies. Refer to their respective documentation for more details.

Upgrading to VxFAS HA Agents 5.0

If Veritas TimeFinder 4.0 HA Agents or VxFAS 4.0 HA Agents is already installed, you can upgrade to this release.

To upgrade to this release from VxTF 4.0 or VxFAS 4.0 HA Agents:

The resource configuration for VxTF 4.0 or VxFAS 4.0 HA Agents can be reused with this release of VxFAS HA Agents.

- 1 Perform pending recovery, if any, for VxTF or VxFAS.
- 2 Uninstall old `VRTStfag` package if upgrading from VxTF 4.0 HA agent or `VRTSfasag` package if upgrading from VxFAS 4.0 HA agents.
- 3 Offline all VCS service groups.
- 4 Upgrade to VCS version 4.1.
- 5 Add the new `VRTSfasag` 5.0 package.
- 6 Online the service groups.

Configuring VxFAS HA agents

To define the VCS resource types required by VxFAS HA feature, run the `vxtfvcsconf` script (without any arguments) from any system in the VCS cluster. See “[Activating VxFAS in the VCS environment](#)” on page 15 for more details.

To get an overview read “[Resources and Service groups for VxFAS HA feature](#)” on page 12.

For a new installation, see “[Guidelines for creating service groups and resources](#)” on page 15.

To upgrade the existing VxFAS configuration to this new HA feature, see “[Upgrading to VxFAS HA Agents 5.0](#)” on page 10.

Modifying the agents and their resources

You can use the VCS Cluster Manager GUI, or enter VCS commands from the command line, to dynamically modify the configuration of resources managed by an agent. Because of its ease of use, it is preferable to use the Veritas Cluster Server GUI to administer VxFAS resources. See the *Veritas Cluster Server Installation Guide* for instructions on how to install the VCS Cluster Manager GUI, and the *Veritas Cluster Server User's Guide* for details on how to perform VCS administrative tasks using this GUI.

You can also edit the `main.cf` configuration file directly, but this is *not* recommended. An example `main.cf` file is located in the `/etc/VRTSvcs/conf/sample_vxtf` directory and under “[Example main.cf file](#)” on page 46.

Resources and Service groups for VxFAS HA feature

Creating a high availability configuration for the VxFAS through VCS requires various resources types, resources, and service groups. The VCS resource types for the VxFAS HA are implemented by the following agents:

- VxSymDevGrp Agent
- VxSymLog Agent
- VxSymRecover Agent
- DiskGroup Agent (bundled VCS agent)
- Mount Agent (bundled VCS agent)

These agents are described in [Chapter 3. “VxFAS Agent Definitions” on page 37](#).

The installation procedure for VxFAS (see “[Activating VxFAS in the VCS environment](#)” on page 15) creates the VCS resource types VxSymDevGrp, VxSymLog, and VxSymRecover. The DiskGroup and Mount resource types are bundled with the VCS. After installation, you can create a VCS service group, of any name, and add the resource instances of the above resource types. These resource instances in turn provide the high availability functionality for VxFAS. All these resource instances (except, in some cases, the Mount resource instance) must be running for any VxFAS operation to succeed in an HA environment.

Depending upon the application requirement, you may want to failover multiple related disk groups to an another system in the cluster. In this case, it is advisable to create a single VCS service group comprised of multiple instances of VxSymDevGrp, multiple instances of DiskGroup, multiple instances of Mount, but only a single instance of VxSymLog and a single instance of VxSymRecover. The VxFAS operations, however, can be carried out on only one disk group at a time.

The VxFAS HA agents do not support simultaneous operations from multiple systems on different disk groups in the same service group. In VCS terminology, it does not support a parallel service group. It also does not support operations on parallel resources such as a CVM shared disk group.

Resource dependencies

Dependencies between resources specify the order in which the resources within a service group come online and go offline. The various resources must be linked so that they meet the following dependencies:

- VxSymRecover resource depends on VxSymLog resource
- Mount resource depends on its associated DiskGroup resource
- All DiskGroup resources depend on VxSymLog resource
- VxSymLog resource depends on all VxSymDevGrp resources

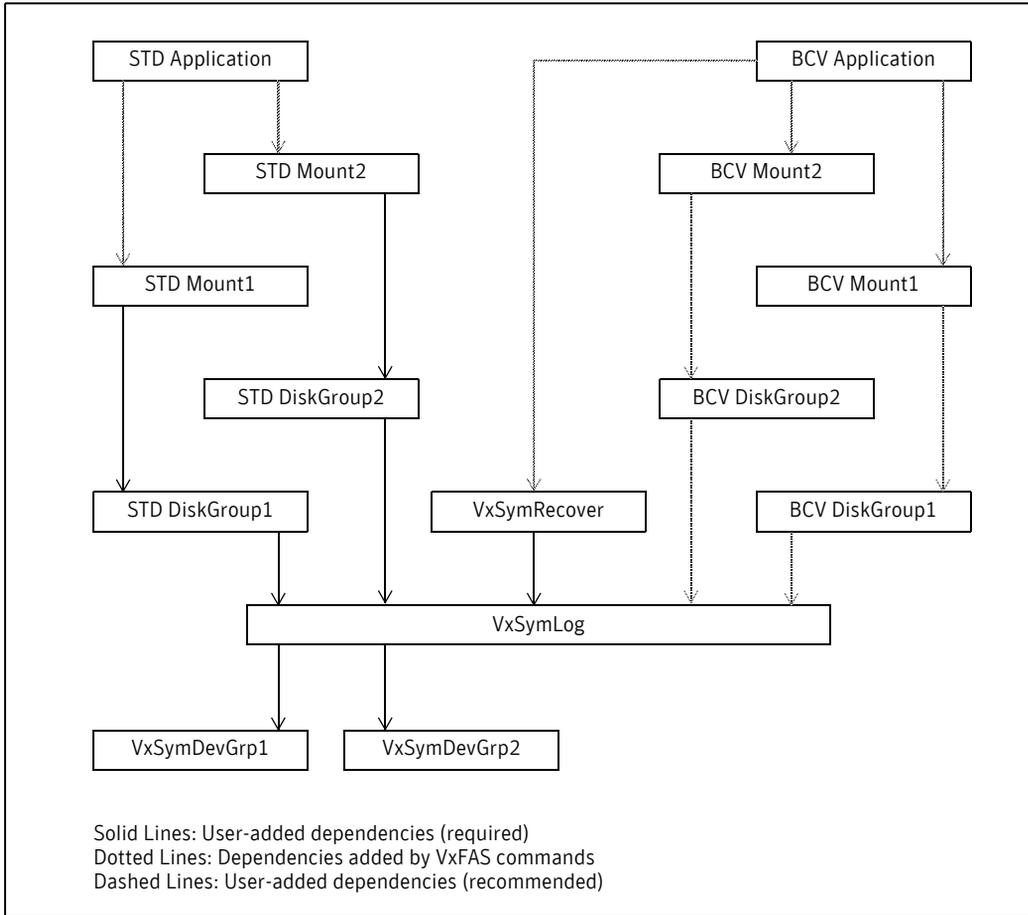
You must define these dependencies using the `hares -link` command (see the `hares(1M)` manual page). The dependencies are shown in the figure “[VxFAS Example Service Group Resource Dependencies](#)” on page 14.

The resources named “STD” are user-created. The resources named “BCV” (except the BCV Application resource) are created by the VxFAS commands. Otherwise, the resource types are the same.

An application, such as a database, running on a file system mount point or disk group, may have its own VCS resources. In this case, depending on whether the application is an STD or a BCV application, you can link it manually as shown in the figure.

This application dependency is not mandatory for the VxFAS HA Agents, but other dependencies are required.

Figure 1-1 VxFAS Example Service Group Resource Dependencies



Activating VxFAS in the VCS environment

Before operating the VxFAS in a VCS Environment with a Symmetrix disk array, follow the procedure in “[Symmetrix-specific setup](#)” on page 23.

Use the `/etc/vx/emc.d/bin/vxtfvcsconf` script to dynamically add VxFAS-specific resource types (listed under “[Resources and Service groups for VxFAS HA feature](#)” on page 12) to a running VCS system. The `vxtfvcsconf` script is a non-interactive utility that adds VxFAS-specific resource types in the VCS environment.

Note: Run this script only once from any node in the cluster.

VCS must be started and its configuration must be read-write (see the `haconf(1M)` manual page for more information) before you run the `vxtfvcsconf` script.

All the VxFAS-specific resource types have the attribute `NumThreads` set to 1 and `OnlineTimeout` set to 86400. Modifying these values can result in unpredictable behavior.

Guidelines for creating service groups and resources

Follow the guidelines below when creating resources for the VxFAS HA feature. Use the Cluster Manager GUI, VCS commands, or the `vxtfmkvcres` utility located under `/etc/vx/emc.d/bin` directory to create service groups and VxFAS related resources.

To upgrade an existing VxFAS configuration to this new HA feature, follow the steps under “[VxFAS HA operational notes](#)” on page 26.

Note: VCS configuration must be read-write before creating any service group and resources.

It is advisable to first dump the configuration to disk and then make it read-only after you are done creating service groups and resources.

Creating a service group

- The service group must have the Parallel attribute cleared (this is the default setting).
- The service group must have the SystemList attribute populated.
- The service group must have the AutoStartList attribute populated.
- The service group must have the AutoFailOver, AutoRestart, and AutoStart attributes set (this is the default setting).

To create a service group, with two systems, using the VCS commands, enter:

```
# hagr -add my_service_group
# hagr -modify my_service_group SystemList system0 0 system1 1
# hagr -modify my_service_group AutoStartList system0 system1
```

To create a service group using the `vxtfmkvsres` utility:

```
# vxtfmkvsres -s my_service_group -l system0,system1
```

Note: Specify systems to the `vxtfmkvsres -l` command in the order of their priority. The system with high priority (lower number) must come before system with lower priority (high number).

Creating a resource instance for a VxSymDevGrp resource

- The VxSymDevGrp instance must have the SetupInfo and SnapObjType attributes unset. Their values are updated from within the VxFAS operations. These attributes are documented as the non-modifiable attributes in the respective agent documentation.
- The VxSymDevGrp instance can be any name that is unique in a given cluster.
- The VxSymDevGrp instance must have its DgName attribute set to the associated VxVM disk group.
- You must put the VxSymDevGrp instance online manually. This is required only when you create the resource instance.
- You must link the VxSymDevGrp instance with the VxSymLog instance as shown in the figure in the section “[Resource dependencies](#)” on page 13.

To create a Resource Instance for a VxSymDevGrp Resource using the VCS commands, enter:

```
# hares -add my_devg VxSymDevGrp my_service_group
# hares -modify my_devg DgName my_real_dg
# hares -link my_log my_devg
# hares -modify my_devg Enabled 1
# hares -online my_devg -sys system0
```

Note: You must create the VxSymLog instance before the VxSymDevGrp instance for the `hares -link` command to succeed.

Alternatively:

```
# vxtfmkvcsres -s my_service_group -t VxSymDevGrp -r my_devg \
-n my_real_dg
# hares -modify my_devg Enabled 1
# hares -online my_devg -sys system0
```

Note: When a VxVM disk group is split into two disk groups, you must delete the VxSymDevGrp type of resource corresponding to that disk group. You should create the VxSymDevGrp type of resources for the newly created disk groups after the disk group split, using the `vxtfmkvcsres` utility as shown above for each disk group.

Creating a resource instance for a VxSymLog resource

- The VxSymLog instance must have the LogDiskList attribute set. The value of LogDiskList attribute is comma separated list of UID attribute values of the Symmetrix devices. (see “[Modifying the VxSymLog LogDiskList attribute](#)” on page 24 for details). The UIDs of the devices can be obtained from *Veritas Enterprise Administrator* GUI. An example value of LogDiskList attribute is:

```
LogDiskList=EMC000134984068018F,EMC000134984068018E
```

- All the log disks must be accessible from all the systems specified in the SystemList attribute of the service group.
- The log disks must have a Device Capacity of at least three megabytes.
- There is no restriction on the Device Configuration (2-Way Mir, for example) of a log device except that it cannot be a CKD or a VCMDDB type device. It also *cannot* be a Gatekeeper device (GK).
- You must ensure that there is only one instance of the VxSymLog resource per service group.
- The VxSymLog instance must have the CurrLogDisk attribute unset.
- You must put the VxSymLog instance online manually. This is required only when you create the resource instance.
- You must link the VxSymLog instance with all the VxSymDevGrp instances as shown in the figure in the section “[Resource dependencies](#)” on page 13. See also “[Creating a resource instance for a VxSymDevGrp resource](#)” on page 17.

Note: Be sure that the Symmetrix disks defined as log disks are not used in any other application.

To create a Resource Instance for a VxSymLog Resource using the VCS commands, enter:

```
# hares -add my_log VxSymLog my_service_group
# hares -modify my_log LogDiskList \
    EMC000134984068018F,EMC000134984068018E
# hares -modify my_log Enabled 1
# hares -online my_log -sys system0
```

Alternatively:

```
# vxxtfmkvcsres -s my_service_group -t VxSymLog -r my_log \
    -i 000134984068 -d 018F,018E
# hares -modify my_log Enabled 1
# hares -online my_log -sys system0
```

Creating a resource instance for a DiskGroup resource

- The DiskGroup instance must have the StartVolumes attribute set (this is the default setting).
- The DiskGroup instance must have the StopVolumes attribute set (this is the default setting).
- The DiskGroup instance can be any name that is unique in a cluster.
- The DiskGroup instance must have its DiskGroup attribute set to the associated Veritas Volume Manager™ (VxVM) disk group name.
- You must link the DiskGroup instance with the VxSymLog instance as shown in the figure in the section “[Resource dependencies](#)” on page 13.

To create a Resource Instance for a DiskGroup Resource using the VCS commands, enter:

```
# hares -add my_dg DiskGroup my_service_group
# hares -modify my_dg DiskGroup my_real_dg
# hares -link my_dg my_log
# hares -modify my_dg Enabled 1
```

Alternatively:

```
# vxtfmkvsres -s my_service_group -t DiskGroup -r my_dg \
-n my_real_dg
# hares -modify my_dg Enabled 1
```

Creating a resource instance for a mount resource

- The Mount instance must have the MountPoint, BlockDevice, and FSType attributes set.

Note: VxFAS configurations support only the Veritas File System (VxFS) as the FSType for primary side (STD) mounts.

- The Mount instance must have the MountPoint attribute set to the full path name for the mount.
- The Mount instance must have the BlockDevice attribute set to the full path name of the device to be mounted.
- Set the Mount instance MountOpt attribute to the preferred mount command options. Leave this blank if you want the default mount options.
- Set the Mount instance FsckOpt attribute to the preferred fsck command options. You must put either the `-y` or `-n` option for `fsck` in this field. For more information see the *Veritas Cluster Server User's Guide*.
- You must link the Mount instance with the corresponding DiskGroup instance as shown in the figure in the section "[Resource dependencies](#)" on page 13.

To create a Resource Instance for a Mount Resource using the VCS commands, enter:

```
# hares -add my_mount Mount my_service_group
# hares -modify my_mount MountPoint "/my_dir"
# hares -modify my_mount BlockDevice \
    "/dev/vx/dsk/my_real_dg/my_vol"
# hares -modify my_mount FSType vxfs
# hares -modify my_mount FsckOpt %"-o full -y"
# hares -link my_mount my_dg
# hares -modify my_mount Enabled 1
```

Alternatively:

```
# vxtfmkvcsres -s my_service_group -t Mount -r my_mount \
    -m "/my_dir" -d "/dev/vx/dsk/my_real_dg/my_vol" -F vxfs \
    -f "-o full -y"
# hares -modify my_mount Enabled 1
```

Creating a resource instance for a VxSymRecover resource

- Be sure that there is only one instance of the VxSymRecover resource per service group.
- You must put the VxSymRecover instance online manually. This is required only when you create the resource instance.
- You must link the VxSymRecover instance with the VxSymLog instance as shown in the figure in the section “[Resource dependencies](#)” on page 13.

To create a Resource Instance for a VxSymRecover Resource using the VCS commands, enter:

```
# hares -add my_recover VxSymRecover my_service_group
# hares -link my_recover my_log
# hares -modify my_recover Enabled 1
# hares -online my_recover -sys system0
```

Alternatively:

```
# vxtfmkvcres -s my_service_group -t VxSymRecover -r my_recover
# hares -modify my_recover Enabled 1
# hares -online my_recover -sys system0
```


Operational Notes

Symmetrix-specific setup

When setting up a VCS cluster using Symmetrix as shared storage, follow these procedures on each node in the cluster. If not followed correctly, a new system added to a cluster will not have OS nodes for the NR devices, and a subsequent failover to a node involving NR devices will not succeed.

Set up a VCS cluster using Symmetrix as shared storage:

- 1 Run the following commands:

```
# drvconfig  
# disks  
# devlinks
```

- 2 Run the following Veritas Volume Manager commands:

```
# vxconfigd  
# vxctl enable
```

- 3 Run these SYMCLI commands:

```
# symcfg discover  
# sympd list
```

The output of the `sympd list` command should not show Device Status (Sts field under the heading `Device` in the command output) as NR (not ready) for any Physical Device. Consult the Symmetrix documentation to change the Status of devices from NR (not ready) to RW (read-write).

Note: If the device was NR when the system was booted, and if the device is changed to RW after the system comes up, run the `vxctl enable` command on all nodes in the cluster (see the `vxctl(1M)` manual page for more information).

Modifying the VxSymLog LogDiskList attribute

The VxSymLog resource type (see “VxSymLog agent” on page 40) has the attribute LogDiskList that you must maintain. LogDiskList is a string-vector type of attribute (see *Veritas Cluster Server User’s Guide* for information about VCS Attribute Data, Type, and Dimension). To modify the value of LogDiskList, you use the `hares -modify` command (see *Veritas Cluster Server User’s Guide* and the `hares(1M)` manual page). Below are some examples of modifying LogDiskList.

- Obtain the value of UID attribute of the required Symmetrix device from the Veritas Enterprise Administrator GUI. An example UID is EMC000134984068027F.

- If the VxSymLog resource name in a particular service group is “vxsymlog,” to define the initial log disk as a disk with UID “EMC000134984068027F” enter:

```
# hares -modify vxsymlog LogDiskList EMC000134984068027F
```

This sets LogDiskList to:

```
LogDiskList = { EMC000134984068027F }
```

- To add a log disk with UID “EMC000134478589018A” enter:

```
# hares -modify vxsymlog LogDiskList -add  
EMC000134478589018A
```

This sets LogDiskList to:

```
LogDiskList = { EMC000134984068027F, EMC00013447858918A  
}
```

- To add a log disk using `vxtfmkvcsres` script, you can use the following command:

```
# vxtfmkvcsres -s <service group> -t VxSymLog -r vxsymlog  
-a \  
-i 000134984068 -d 027F
```

- To delete a log disk with UID “EMC000134984068027F” from the LogDiskList, enter:

```
# hares -modify vxsymlog LogDiskList -update  
EMC000134478589018A
```

This sets LogDiskList to:

```
LogDiskList = { EMC000134478589018A }
```

- To delete all the log disks defined so you can re-define them, enter:

```
# hares -modify vxsymlog LogDiskList -delete -keys
```

Note: Running the `hares -modify` command without a keyword (`add`, `delete`, or `update`) overwrites the earlier value of an attribute as in the first example above.

VxFAS HA operational notes

The following are notes on the operation of the VxFAS HA feature.

Note: The StorageAgent should be in a RUNNING state on the nodes in the cluster before you run any VxFAS command.

VxFAS command-specific issues

Execution modes

The VxFAS commands that support the HA agents can run in HA or non-HA mode. The pre-conditions for running a VxFAS command in HA mode are:

- VCS engine is running on the system
- The VxSymDevGrp resource corresponding to the VxVM disk group specified on the command line (using the `-g` option) is online

If the pre-conditions are not satisfied, a VxFAS command runs in non-HA mode. The command prints a message when running in non-HA mode. Because of this behavior, you must run the entire life cycle of VxFAS configurations either in HA or non-HA mode. The life cycle being:

```
vxsymsetup attach -> vxsymmir -> vxsymsplit -> vxsymmir ->  
vxsymsplit -> vxsymsetup detach
```

or

```
vxsymsetup attach -> vxsymmir -> vxsymsplit -> vxsymrestore ->  
vxsymsplit -> vxsymsetup detach
```

Command fails during recovery

If a node goes down while a VxFAS command is running, the same command is re-executed on the failover host as a part of the recovery.

If this command fails on the other host during the recovery, a message is logged in the VCS logs (in the `VxSymLog_A.log` or in `VxSymRecover_A.log` file under the `/var/VRTSvcs/log` directory) along with the error code and output of the failing command:

```
An error occurred during recovery
```

At this point the service group fails over to the next available node in its SystemList attribute and tries to recover the failed command on this host. If the problem persists on the last node in the SystemList attribute, you must intervene manually.

To recover from this situation, determine the cause of the failure, then put the service group online using the following commands:

```
# hagrp -clear group_name  
# hagrp -online group_name -sys system_name
```

Logging-specific issues

A VxFAS command logs its progress to a log disk. The log disk is reserved as defined in the LogDiskList attribute. The command assigns the log disk name to the CurrLogDisk attribute and starts logging.

Node crashes before assigning a log disk

A node may crash after the log disk is reserved but before its name is assigned to the CurrLogDisk attribute. This leaves the log disk reserved forever, making it unusable by other VxFAS commands.

To determine if there are any permanently reserved log disks, use the vxtfunresv utility located in /etc/vx/emc.d/bin directory:

```
# vxtfunresv  
vxtfunresv: INFO: Following device(s) appear to be reserved, but  
not being used as log device for VxTF EMC000134478589018A  
Unreserve the log device(s)? [y/n]:
```

To unreserve the listed log disks, input **y** at the prompt or run `vxtfunresv -y`.

Node crashes after assigning a log disk

A node may crash after adding the log disk name to the CurrLogDisk attribute but before logging actually starts. In this case, the command cannot be recovered on the other host and a message is logged in the VCS logs (in the VxSymLog_A.log or in VxSymRecover_A.log file under the /var/VRTSvcs/log directory):

```
Re-run the previous command on this node
```

In this case, recover the command run on the previous host by running the vxsymrecover command or from Veritas Enterprise Administrator GUI.

Recovering a terminated VxFAS command

If you terminate the `vxpal` process running StorageAgent (`vxpal -a StorageAgent`) using the `kill` command (see the `kill(1)` while a VxFAS operation was in progress, it may leave an unprocessed log causing subsequent commands to fail.

To resolve this condition, you must either take the service group offline, restart the `StorageAgent` process and put it back online on the same system, or switch the service group to a different system in the cluster. This processes the log and completes the execution of a VxFAS command.

To take the service group offline and then put it back online on the same system (`system_0`), enter:

```
# hagr -offline my_service_group -sys system_0
# hagr -online my_service_group -sys system_0
```

Note: The `StorageAgent` must be in `RUNNING` state before the service group is put online on the system. To check the state of `StorageAgent`, enter the following command:

```
# /opt/VRTSobc/pa133/bin/vxpalctrl -a StorageAgent -c
status
```

To switch the service group to a different system in the cluster (`system_1`), enter:

```
# hagr -switch my_service_group -to system_1
```

VxVM-specific issues

Mount Resource Failure Due to Disks in an Error State

If one or more disks in a disk group enter an error state and disable the volumes in that disk group, the Mount resources for the volumes will fail. In this case, you may need to online the disks using the `vxdisk online` or `vxctl enable` command. This problem occurs in only a few situations, such as when the `vxsysplit` command crashes and fails over.

VCS-specific issues

VCS 4.1 and the sysname file

If the node on which VCS is running uses the domain-qualified system name, that is, the name contains periods, VCS cannot use this name as a system name in its configuration. In this case, you can change the system name using the `uname -s` command, or create the file `/etc/VRTSvcs/conf/sysname` containing a system name without periods (this can be obtained using the `uname -n` command).

If you create the `sysname` file, be sure to create or edit this file after shutting down VCS on that node.

TypeDependencies attribute

VCS provides a method for establishing dependencies between resources according to the resource type. For example, to configure all resources of type Mount to depend on all resources of type DiskGroup, define the TypeDependencies service group attribute as:

```
TypeDependencies = { Mount, DiskGroup }
```

Note: Due to complex resource dependencies in the VxFAS HA agents, do not set this attribute for any service group containing VxFAS-specific resource types as the results are unpredictable. Setting the TypeDependencies to the value shown in the above example is dangerous as it makes the STD Mount resources implicitly dependent on the BCV DiskGroup resources.

VCS configuration modes and VxFAS commands

VxFAS commands modify the VCS configuration, so the configuration mode must be Read/Write before they start running.

If the configuration is in Read-Only mode, the VxFAS command can make it Read/Write when it starts and restore it back to Read-Only before it completes. This can be done by running the VxFAS CLI with the `-n` option or by checking the **Make VCS Configuration Read-write option** in the GUI.

Caution: It is important to keep the VCS configuration Read-Only when the VxFAS commands are not running.

VCS configuration modes and recovery

VxFAS commands running on the failover host during a recovery also require that the VCS Configuration is Read/Write. Because commands on the failover host run non-interactively, the configuration is made Read/Write and restored back to Read-Only internally by the command.

It is good practice to check the configuration mode after the failover completes and the cluster is stable. Set the VCS configuration mode to Read-Only if the recovering command could not do so:

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

Namespaces for VxFAS controlled objects

VCS requires unique resource names throughout the cluster. Similarly, the object names pointed to by these resources must be unique. All of the following must be unique in the cluster:

- Disk group name in the DiskGroup attribute of a DiskGroup resource and DgName attribute of a VxSymDevGrp resource
- Mount point name in the MountPoint attribute of a Mount resource

BCV resources

VxFAS commands create the resources of type DiskGroup and Mount corresponding to the BCV disk group and BCV mount point respectively. The Critical attribute of these resources is set to zero so that their failure does not initiate a failover to the other host.

BCV resources and resource names

Typically, a BCV resource name is the same as that of a disk group or mount point that the BCV resource controls. If this name is already used by another resource in the cluster, the command chooses another name for the BCV resource to avoid a conflict. For example, if the resource name “BCVmy_dg” is already in use, the command selects “BCVmy_dg1” as the resource name.

Application-specific resources dependent on BCV resources

You can manually add the application-specific resources that are dependent on the BCV resources only after running the VxFAS `vxsysmsplit` command. This command creates the BCV resources internally.

You must subsequently delete or offline application-specific resources before running the `vxsymmir` or `vxsymrestore` commands. These two commands delete the BCV resources created by the `vxsysmsplit` command. You can add or online the BCV resources again after the next `vxsysmsplit` command. See the `vxsymmir(1M)`, `vxsymrestore(1M)`, and `vxsysmsplit(1M)` manual pages for more information.

Application-specific resources dependent on STD resources

You can manually add the application-specific resources that are dependent on the STD resources, for example, a database application resource. You must delete or offline these application-specific resource before running the `vxsymrestore` command as this command offlines and disables the STD resources. After the `vxsymrestore` command completes, you can add or online application-specific STD resources again.

Cluster crash recovery

If all nodes in a cluster go down simultaneously while the VCS configuration is Read/Write, when they come up they enter into the STALE_ADMIN_WAIT state. To recover a crashed VxFAS command in this case, you must manually run the `vxtfvcschk` script located in `/etc/vx/emc.d/bin` directory. Run this script on the node where you intend to start VCS. Run `vxtfvcschk` without any arguments and follow the instructions displayed by the script. If nodes in a cluster go down at different times, run the `vxtfvcschk` script on the system which went down last.

Note: You must run the `vxtfvcschk` script before running the `hasys -force system_name` command.

EMC PowerPath-specific issues

VxFAS HA agents support PowerPath devices with the following restrictions:

- PowerPath must be installed on all the nodes in the cluster or none of the nodes in the cluster.
- The PowerPath driver version on all cluster nodes must be same.
- Symmetrix device groups controlled by VxFAS HA agents cannot contain a mix of PowerPath devices and non-PowerPath devices.
- If the PowerPath driver name changes, a Symmetrix device group cannot have a mix of PowerPath devices with the old and new names.

Re-using devices from other device groups

This version of the VxFAS supports HA and non-HA VxFAS device groups simultaneously. The device information of the non-HA VxFAS device groups is maintained in a host-specific file named `symapi_db.bin`. The device information of the HA device groups is maintained in a host-specific `symapi_db.bin` file and a cluster configuration file. There may also be non-VxFAS device group (that is, a device group that has names that do not begin with “vxvm_”) information maintained, along with VxFAS device group information, in the host-specific `symapi_db.bin` file.

The VxFAS running in HA mode does not allow re-use of a particular device in a different device group if that device is used in:

- Any non-VxFAS device group defined on the current host
- Any non-HA VxFAS device group defined on the current host
- Any HA VxFAS device group (irrespective of its state) defined in the cluster

The VxFAS running in non-HA mode cannot perform the last check above; you must do this manually. However, if you try to re-use a device already used in a non-HA VxFAS device group, or in a non-VxFAS device group defined on another host, it cannot be detected and a command will succeed. Be careful to avoid this situation.

Note: If you try to create a VCS resource for a non-VxFAS device group, the results are unpredictable. So the category “HA non-VxFAS device group” cannot exist, and the term “HA device group” always means “HA VxFAS device group.”

Re-use examples

Consider an example where an HA device group is online on a host named H1. Assume that a non-HA VxFAS device group, defined on a node named H2 in the same cluster, has re-used a device from the HA device group. If you try to switch the service group corresponding to the HA device group to host H2, the `VxSymDevGrp` resource for the device group faults due to the device group membership conflict. The service group will fail over to the next available host in the `SystemList` attribute of the service group.

In the following examples, assume a VxFAS command running in HA mode and trying to re-use a device fails over and that the cluster has two nodes: H1 and H2.

Case 1

The `vxsymsetup attach` command is run on host H1 and tries to re-use a device from a non-VxFAS device group on host H2. VxFAS allow this as the re-use cannot be detected.

If the command fails over to host H2 right after it was logged, the recovery of the command fails because the re-use is now detected on host H2. The command will succeed on any host other than H2. However, if the command fails over to host H2 after it was logged, and the device was re-used on H1, it will pull the conflicting device to the new device group on H2 and the recovery will be complete. The log message to that extent will be written to the VxSymDevGrp resource's log file.

Case 2

The `vxsymsetup attach` command is run on host H1 and tries to re-use a device from a non-HA VxFAS device group defined on host H2. VxFAS will allow this as the re-use cannot be detected.

If the command fails over to host H2 immediately after it was logged, the recovery of a command will fail as the re-use is now detected on host H2. The command will succeed on any host other than H2. However, if the command fails over to host H2 after the device was re-used on H1, the recovery of a command will fail to indicate that this conflicts with a valid VxFAS device group on host H2. You must take a corrective action, such as removing the re-used device from the non-HA VxFAS device group defined on host H2. The command will succeed on any host other than H2.

Case 3

The `vxsymsetup attach` command is run on host H1 and tries to re-use a device from either a non-HA VxFAS or a non-VxFAS device group defined on host H1. Typically this command will fail as the re-use is detectable. If the command fails over to host H2, it will always succeed as the re-use cannot be detected.

Note: It is advisable to define both non-HA VxFAS and non-VxFAS device groups on hosts other than those in the VCS cluster.

Converting non-HA setup to HA

You can convert an existing non-HA setup under VCS control and use the high availability feature of the VxFAS commands.

Convert a non-HA setup to HA:

- 1 Using the VEA GUI, perform a rescan of the arrays connected to the nodes in the cluster.

Note: Set the path environment variable to include `/etc/vx/emc.d/bin` and `/opt/VRTS/bin` directories. The StorageAgent must be in the RUNNING state before you run these commands. All the following commands must be run on that node in the cluster on which the VxFAS commands were running in non-HA mode.

- 2 Create a service group. For example, if your host names are “node1” and “node2”:

```
# vxtfmkvcsres -s mysg -l node1,node2
```
- 3 Add VxSymDevGrp type of resources for each of the STD disk groups to put under HA control using utility `vxtfmkvcsres`. For example, if the STD disk group name is “mydg” and you want the VxSymDevGrp resource name “mydevg,” enter:

```
# vxtfmkvcsres -s mysg -t VxSymDevGrp -r mydevg -n mydg
```
- 4 Add a VxSymLog type of resource to the service group. For example, to add a VxSymLog resource with name “mylog” and log devices 027F and 018A from symmetrix 000134984068, enter:

```
# vxtfmkvcsres -s mysg -t VxSymLog -r mylog -i  
000134984068 \  
-d 0027,018A
```
- 5 Add a VxSymRecover type of resource to the service group. To add VxSymRecover resource with name “myrec,” enter:

```
# vxtfmkvcsres -s mysg -t VxSymRecover -r myrec
```
- 6 For each disk group to put under HA control, add a DiskGroup type of resource to the service group. If you have already split BCV disk groups, add resources for those disk groups:

```
# vxtfmkvcsres -s mysg -t DiskGroup -r mydg -n mydg
```

- 7 For each mount point on the disk groups for which you have created DiskGroup type resources, add Mount type resources to the service group. For example, if you have mounted `/dev/vx/dsk/mydg/myvol` on `/mnt/myvol`, enter:

```
# vxdfmkvcsres -s mysg -t Mount -r mnt_myvol -m  
/mnt/myvol \  
-F vxfs -d /dev/vx/dsk/mydg/myvol -f "-y"
```

- 8 Although it is not required to create Volume type resources for VxFAS HA functionality, you can create them using the `hares` command or by other means provided by VCS. See the *Veritas Cluster Server User's Guide* for information on how to create resources.
- 9 Check that the resource dependencies are as described in the section "[Resource dependencies](#)" on page 13.
- 10 Unmount the file systems and deport all the VxVM disk groups for which you have created the DiskGroup type of resources.
- 11 Enable all the resources in the service group.

```
# hagr -enableresources mysg
```

- 12 Online the service group on any of the nodes in the cluster.

```
# hagr -online mysg -sys node1
```

After the service group is online on the node, you can perform the VxFAS operations on the disk groups that are under VCS control in HA mode from that node.

VxFAS Agent Definitions

This chapter describes the VxFAS HA agents and the bundled VCS agents required to support high availability functionality.

- [VxSymDevGrp agent](#)
- [VxSymLog agent](#)
- [VxSymRecover agent](#)

The HA feature requires these VCS bundled agents:

- [DiskGroup agent](#)
- [Mount agent](#)

There is also an example `main.cf` configuration file.

- [Example main.cf file](#)

VxSymDevGrp agent

Table 3-1

Description	Creates, deletes, and monitors a Symmetrix device group.	
Entry Points	<ul style="list-style-type: none"> ■ Online—Creates a Symmetrix device group of the required configuration. ■ Offline—Deletes the Symmetrix device group. ■ Monitor—Determines if the Symmetrix device group is visible to the host. ■ Clean—A null operation. 	
Required Attributes	Type and Dimension	Definition
DgName	string-scalar	Name of the associated Volume Manager disk group.
Non-modifiable Attributes	Type and Dimension	Definition
SetupInfo	string-vector	Information about the device group, such as the device group type and the list of devices in device group.
SnapObjType	string-scalar	Type name of an associated Veritas Enterprise Administrator (VEA) GUI object.
OnlineHost	string-scalar	Host name where the resource is online.

Type definition

```
type VxSymDevGrp (  
    static int NumThreads = 1  
    static int OnlineTimeout = 86400  
    static str ArgList[] = { OnlineHost, DgName, SnapObjType,  
SetupInfo}  
    NameRule = ""  
    str DgName  
    str SetupInfo[]  
    str SnapObjType  
    str OnlineHost  
)
```

Sample configuration

```
VxSymDevGrp demo01_dg (  
    DgName = demo01_dg  
)  
  
VxSymDevGrp demo02_dg (  
    DgName = demo02_dg  
    SetupInfo = { "DevName=vxvm_demo02_dg" }  
    SnapObjType = emc_symmetrix_snapshot_service  
    OnlineHost = system0  
)
```

VxSymLog agent

Table 3-2

Description	Initiates recovery of a crashed VxFAS command.	
Entry Points	<ul style="list-style-type: none"> ■ Online—Initiates recovery of a crashed <code>restore</code> command. ■ Offline—A Null operation. ■ Monitor—Reports whether the VxSymLog resource is online or offline. ■ Clean—A null operation. 	
Required Attributes	Type and Dimension	Definition
LogDiskList	string-keylist	List of UIDs of the allocated Symmetrix log disks.
Non-modifiable Attributes	Type and Dimension	Definition
CurrLogDisk	string-scalar	UID of the current log disk.
OnlineHost	string-scalar	Host name where the resource is online.

Type definition

```
type VxSymLog (  
    static int NumThreads = 1  
    static int OnlineTimeout = 86400  
    static str ArgList[] = { OnlineHost, CurrLogDisk, LogDiskList }  
    NameRule = ""  
    str LogDiskList[]  
    str CurrLogDisk  
    str OnlineHost  
)
```

Sample configuration

```
VxSymLog vxsymlog (  
    LogDiskList = { EMC000134984068018F, EMC000134984068018E }  
    CurrLogDisk = { EMC000134984068018E }  
    OnlineHost = system0  
)
```

VxSymRecover agent

Table 3-3

Description	Recovers a crashed VxFAS command.	
Entry Points	<ul style="list-style-type: none">■ Online—Reads the log and recovers the crashed command.■ Offline—A null operation.■ Monitor—Reports if the VxSymRecover resource is online or offline.■ Clean—A null operation.	
Non-modifiable Attributes	Type and Dimension	Definition
OnlineHost	string-scalar	Host name where the resource is online.

Type definition

```
type VxSymRecover (  
    static int NumThreads = 1  
    static int OnlineTimeout = 86400  
    static str ArgList[] = { OnlineHost }  
    NameRule = ""  
    str OnlineHost  
)
```

Sample configuration

```
VxSymRecover vxsymrecover (  
    OnlineHost = system1  
)
```

DiskGroup agent

Table 3-4

Description	Brings online, takes offline, and monitors a Veritas Volume Manager disk group.	
Entry Points	<ul style="list-style-type: none"> ■ Online—This script imports the disk group using the <code>vxldg</code> command. ■ Offline—This script deports the disk group using the <code>vxldg</code> command. ■ Monitor—This agent determines if the disk group is online or offline using the <code>vxldg</code> command. If the disk group was imported with <code>noautoimport=off</code>, and if the group is not frozen, the group to which the disk group belongs is taken offline. 	
Required Attribute	Type and Dimension	Definition
DiskGroup	string-scalar	Disk group name.
Optional Attributes	Type and Dimension	Definition
StartVolumes	string-scalar	If value is 1, the DiskGroup <code>online</code> script starts all volumes belonging to that disk group after importing. Default is 1.
StopVolumes	string-scalar	If value is 1, the DiskGroup <code>offline</code> script stops all volumes belonging to that disk group before deporting. Default is 1.

Type definition

```

type DiskGroup (
    static int OnlineRetryLimit = 1
    str DiskGroup
    NameRule = resource.DiskGroup
    static str ArgList[] = { DiskGroup, StartVolumes,
                           StopVolumes,
                           MonitorOnly }
    str StartVolumes = 1
    str StopVolumes = 1
    static int NumThreads = 1
)

```

Sample configuration

```

DiskGroup emcdg (
    DiskGroup = emcdg
)

```

Mount agent

Table 3-5

Description	Brings online, takes offline, and monitors a file system mount point.	
Entry Points	<ul style="list-style-type: none"> ■ Online—Mounts a block device on the directory. If the mount fails, the agent runs <code>fsck</code> on the raw device to remount the block device. ■ Offline—Unmounts the file system. ■ Monitor—Determines if the file system is mounted. Checks mount status using the <code>stat</code> and <code>statvfs</code> commands. 	
Required Attributes	Type and Dimension	Definition
BlockDevice	string-scalar	Block device for mount point.
MountPoint	string-scalar	Directory for mount point.
FSType	string-scalar	File system type, for example, vxfs or ufs.
FsckOpt	string-scalar	Options for <code>fsck</code> command.
Optional Attributes	Type and Dimension	Definition
MountOpt	string-scalar	Options for <code>mount</code> command.
SnapUmount	integer-scalar	If set to 1, VxFS snapshot mounts are unmounted automatically when file system is unmounted. Default is 0 (No).

Type definition

```
type Mount (
    str MountPoint
    str BlockDevice
    str FSType
    str MountOpt
    str FsckOpt
    int SnapUmount = 0
    NameRule = resource.MountPoint
    static str ArgList[] = { MountPoint, BlockDevice, FSType,
MountOpt, FsckOpt, SnapUmount }
)
```

Sample configuration

```
Mount export1 (
    MountPoint= "/export1"
    BlockDevice = "/dev/vx/dsk/emcdg/voll"
    FSType = vxfs
    MountOpt = ro
)
```

Example main.cf file

```
include "types.cf"

cluster vcscluster1 (
    UserNames = { admin = "cDRpdxPmHpzS." }
    CounterInterval = 5
    Factor = { runque = 5, memory = 1, disk = 10, cpu =
25,
                                network = 5 }
    MaxFactor = { runque = 100, memory = 10, disk = 100,
cpu = 100,
                                network = 100 }
)

system system0
system system1
system system2
system system3

snmp vcs (
    TrapList = { 1 = "A new system has joined the VCS
Cluster",
                2 = "An existing system has changed its state",
                3 = "A service group has changed its state",
                4 = "One or more heartbeat links has gone down",
                5 = "An HA service has done a manual restart",
                6 = "An HA service has been manually idled",
                7 = "An HA service has been successfully started"
    }
)

group test_sg1 (
    SystemList = { system1, system0 }
    AutoStartList = { system1, system0 }
)

VxSymDevGrp vxvm_test1_dg (
    DgName = test1_dg
    SetupInfo = { "DevName=vxvm_test1_dg" }
    SnapObjType = emc_symmetrix_snapshot_service
    OnlineHost = system1
)

DiskGroup test1_dg (
    DiskGroup = test1_dg
)

VxSymLog test1_log (
    LogDiskList = { EMC0001349840680120,EMC0001349840680
}
}
```

```
        OnlineHost = system1
    )

VxSymRecover test1_recover (
    OnlineHost = system1
)

test1_dg requires test1_log
test1_recover requires test1_log
test1_log requires vxvm_test1_dg

// resource dependency tree
//
//         group test_sg1
//         {
//         DiskGroup test1_dg
//         {
//             VxSymLog test1_log
//             {
//                 VxSymDevGrp vxvm_test1_dg
//             }
//         }
//         VxSymRecover test1_recover
//         {
//             VxSymLog test1_log
//             {
//                 VxSymDevGrp vxvm_test1_dg
//             }
//         }
//     }

group test_sg2 (
    SystemList = { system0, system1 }
    AutoStartList = { system0, system1 }
)

VxSymDevGrp vxvm_test2_dg (
    DgName = test2_dg
    SetupInfo = { "DvgName=vxvm_test2_dg" }
    SnapObjType = emc_symmetrix_snapshot_service
    OnlineHost = system1
)
```

```
DiskGroup test2_dg (  
    DiskGroup = test2_dg  
)  
  
VxSymLog test2_log (  
    LogDiskList = { EMC000134984068011E }  
    OnlineHost = system1  
)  
  
VxSymRecover test2_recover (  
    OnlineHost = system1  
)
```

test2_dg requires test2_log
test2_log requires vxvm_test2_dg
test2_recover requires test2_log

```
// resource dependency tree  
//  
//          group test_sg2  
//          {  
//          DiskGroup test2_dg  
//          {  
//              VxSymLog test2_log  
//              {  
//                  VxSymDevGrp vxvm_test2_dg  
//              }  
//          }  
//          VxSymRecover test2_recover  
//          {  
//              VxSymLog test2_log  
//              {  
//                  VxSymDevGrp vxvm_test2_dg  
//              }  
//          }  
//          }  
//          }
```

Index

C

configuration files
 types.cf 9

E

Example main.cf 46

F

file systems
 mounting and unmounting 44

T

types.cf file 9

V

VCS Cluster Components 9
VxSymDevGrp Agent 38
VxSymLog Agent 40
VxSymRecover Agent 42

