Veritas™ Cluster Server Bundled Agents Reference Guide

AIX

5.1 Service Pack 1

Symantec

Veritas Cluster Server Bundled Agents Reference 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.

Product version: 5.1 SP1

Document version: 5.1.SP1.3

Legal Notice

Symantec Corporation
350 Ellis Street
Mountain View, CA 94043
http://www.symantec.com

Technical Support

Symantec 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 Symantec to answer your questions in a timely fashion. For example, the Technical Support group works with Product Engineering and Symantec Security Response to provide alerting services and virus definition updates.

Symantec's 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 automatic software upgrades protection
- 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 Symantec's support offerings, you can visit our web site at the following URL:

www.symantec.com/business/support/index.jsp

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.symantec.com/business/support/contact_techsupp_static.jsp

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 Symantec
  - Recent software configuration changes and network changes

Licensing and registration

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

www.symantec.com/business/support/

Customer service

Customer service information is available at the following URL:

www.symantec.com/business/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
- Information about the Symantec Buying Programs
- Advice about Symantec's technical support options
- Nontechnical presales questions
- Issues that are related to CD-ROMs or manuals

Support agreement resources

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

Asia-Pacific and Japan	customercare_apac@symantec.com
Europe, Middle-East, and Africa	semea@symantec.com
North America and Latin America	supportsolutions@symantec.com

Documentation

Product guides are available on the software disc in PDF format. Make sure that you are using the current version of the documentation.The document version appears on page 2 of each guide. The latest product documentation is available on the Symantec website.

http://www.symantec.com/business/support/overview.jsp?pid=15107

Your feedback on product documentation is important to us. Send suggestions for improvements and reports on errors or omissions. Include the title and document version (located on the second page), and chapter and section titles of the text on which you are reporting. Send feedback to:

docs@symantec.com

About Symantec Connect

Symantec Connect is the peer-to-peer technical community site for Symantec's enterprise customers. Participants can connect and share information with other product users, including creating forum posts, articles, videos, downloads, blogs and suggesting ideas, as well as interact with Symantec product teams and Technical Support. Content is rated by the community, and members receive reward points for their contributions.

http://www.symantec.com/connect/storage-management

Contents

Chapter 4 File share agents

Chapter 7 Testing agents

Chapter 1

Introduction

Bundled agents are Veritas Cluster Server (VCS) processes that manage resources of predefined resource types according to commands received from the VCS engine, HAD. You install these agents when you install VCS.

A node has one agent per resource type that monitors all resources of that type. For example, a single IP agent manages all IP resources.

When the agent starts, it obtains the necessary configuration information from VCS. The agent then periodically monitors the resources, and updates VCS with the resource status.

Agents can:

- Bring resources online.
- Take resources offline.
- Monitor resources and report state changes.

For a more detailed overview of how agents work, refer to the *Veritas Cluster Server Administrator's Guide*.

Resources and their attributes

Resources are parts of a system. They are known by their types, for example: a volume, a disk group, or an IP address. VCS includes a set of resource types. Different attributes define these resource types in the types.cf file. Each type has a corresponding agent that controls the resource.

The VCS configuration file, main.cf, contains the values for the resource attributes and has an include directive to the types.cf file.

An attribute's given value configures the resource to function in a specific way. By modifying the value of a resource attribute, you can change the way the VCS agent manages the resource. For example, the IP agent uses the Address attribute to determine the IP address to monitor.

Modifying agents and their resources

Use the Cluster Manager (Java Console), Veritas Operations Manager, or the command line to dynamically modify the configuration of the resources managed by an agent.

VCS enables you to edit the main.cf file directly. To implement these changes, make sure to restart VCS.

See the *Veritas Cluster Server Administrator's Guide* for instructions on how to complete these tasks.

Attributes

Attributes contain data about the cluster, systems, service groups, resources, resource types, and the agent. An attribute has a definition and a value. You change attribute values to configure VCS resources. Attributes are either optional or required, although sometimes attributes that are optional in one configuration might be required in other configurations. Many optional attributes have predefined or default values, which you should change as required.

A variety of internal use only attributes also exist. Do not modify these attributes—modifying them can lead to significant problems for your clusters.

Attributes have type and dimension. Some attribute values can accept numbers, others can accept alphanumeric values or groups of alphanumeric values, while others are simple boolean on/off values.

Table 1-1 Attribute data types

Data Type	Description
string	Enclose strings, which are a sequence of characters, in double quotes ("). Optionally enclose strings in quotes when they begin with a letter, and contains only letters, numbers, dashes (-), and underscores (_). A string can contain double quotes, but the quotes must be immediately preceded by a backslash. In a string, represent a backslash with two backslashes (\\).
integer	Signed integer constants are a sequence of digits from 0 to 9. You can precede them with a dash. They are base 10. Integers cannot exceed the value of a 32-bit signed integer: 2147483647.

Table 1-1 Attribute data types

Data Type	Description
boolean	A boolean is an integer with the possible values of 0 (false) and 1 (true).

Table 1-2 Attribute dimensions

Dimension	Description
scalar	A scalar has only one value. This is the default dimension.
vector	A vector is an ordered list of values. Each value is indexed using a positive integer beginning with zero. A set of brackets ([]) denotes that the dimension is a vector. Find the specified brackets after the attribute name on the attribute definition in the types.cf file.
keylist	A keylist is an unordered list of unique strings.
association	An association is an unordered list of name-value pairs. An equal sign separates each pair. A set of braces ({}) denotes that an attribute is an association. Braces are specified after the attribute name on the attribute definition in the types.cf file, for example: str SnmpConsoles{}.

WPAR-aware agents

Table 1-3 lists the ContainerOpts attribute default values for resource types. Symantec recommends that you do not modify these values.

Table 1-3 ContainerOpts attribute default values for applications and resource types

Resource Type	RunInContainer	PassCInfo
Application	1	0
IP	0	1
IPMultiNICB	0	1
Mount	0	0
Process	1	0
WPAR	0	1

For more information on using WPARs in your VCS environment, refer to the *Veritas Cluster Server Administrator's Guide*.

Enabling debug log messages

To help troubleshoot agent issues, you can enable debug log messages in the agent framework as well as the agents.

To enable agent framework debug log messages:

```
hatype -modify agent_name LogDbg -add DBG_AGDEBUG DBG_AGINFO
DBG_AGTRACE
```

For example:

```
hatype -modify Mount LogDbg -add DBG_AGDEBUG DBG_AGINFO DBG_AGTRACE
```

To enable agent-specific debug log messages:

```
hatype -modify agent_name LogDbg -add debug_log_levels
```

For example:

```
hatype -modify Mount LogDbg -add DBG_1 DBG_2 DBG_3 DBG_4 DBG_5 DBG_6
```

Alternatively, you can also use the following command:

```
hatype -modify Mount LogDbg -add 1 2 3 4 5 6
```

Agent-specific debug log level information is specified in the agent's description. For example, for information about the Mount agent, see "Debug log levels" on page 85.

Chapter 2

Storage agents

This chapter contains:

About the storage agents

Use storage agents to Monitor shared storage.

DiskGroup agent

The DiskGroup agent brings online, takes offline, and monitors Veritas Volume Manager (VxVM) disk groups. This agent uses VxVM commands. You can use this agent to monitor or make disk groups highly available.

When the value of the StartVolumes and StopVolumes attribute is 1, the DiskGroup agent brings the volumes online and takes them offline during the import and deport operations of the disk group.

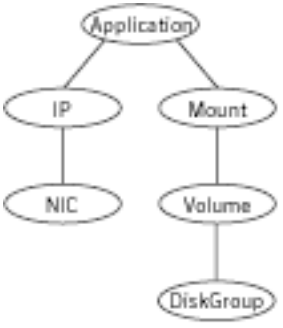
For important information on this agent, refer to:

"DiskGroup agent notes" on page 32

Dependencies

The DiskGroup resource does not depend on any other resources.

Figure 2-1 Sample service group that includes a DiskGroup resource



Agent functions

Online	Imports the disk group using the `vxdg` command.
Offline	Deports the disk group using the `vxdg` command.
Monitor	Determines if the disk group is online or offline using the `vxdg` command. The Monitor function changes the value of the VxVM noautoimport flag from off to on. This action allows VCS to maintain control of importing the disk group. The monitor function uses following command to set the noautoimport flag to on. `# vxdg -g disk_group set autoimport=no`

Clean

Terminates all ongoing resource actions and takes the resource offline—forcibly when necessary.

Info

The DiskGroup info agent function gets information from the Volume Manager and displays the type and free size for the DiskGroup resource.

Initiate the info agent function by setting the InfoInterval timing to a value greater than 0.

In the following example, the info agent function executes every 60 seconds:

```
# haconf -makerw
# hatype -modify DiskGroup InfoInterval 60
```

The command to retrieve information about the DiskType and FreeSize of the DiskGroup resource is:

```
# hares -value diskgroupres ResourceInfo
```

Output includes:

```
DiskType sliced
FreeSize 35354136
```

Action

Different action agent functions follow:

- license.vfd
  Checks for valid Veritas Volume manager license–if one is not found use the vxlicinst utility to install a valid license key.
- disk.vfd
  Checks if all disks in diskgroup are visible on host—if it fails, check if the path to disks exists from the host and check if LUN masking and zoning are set properly.
- udid.vfd
  Checks the UDIDs (unique disk identifiers) of disks on the cluster nodes—if it fails, ensure that the disks that are used for the disk group are the same on all cluster nodes.
- verifyplex.vfd
  Checks if the number of plexes on each site for the Campus Cluster setup are set properly—if it fails, check that the sites, disks, and plexes are set properly for a Campus Cluster setup.
- volinuse
  Checks if open volumes are in use or file systems on volumes that are mounted outside of VCS configuration.

See "High availability fire drill" on page 32.

State definitions

ONLINE	Indicates that the disk group is imported.
OFFLINE	Indicates that the disk group is not imported.
FAULTED	Indicates that the disk group has unexpectedly deported or become disabled.
UNKNOWN	Indicates that a problem exists either with the configuration or the ability to determine the status of the resource. One cause of this state is when I/O fencing is not configured—the cluster level attribute UseFence is not set to "SCSI3" but the Reservation attribute value is "SCSI3".

Attributes

Table 2-1 Required attributes

Required attribute	Description
DiskGroup	Name of the disk group that is configured with Veritas Volume Manager. Type and dimension: string-scalar

Table 2-2 Optional attributes

Optional attributes	Description
StartVolumes	If the value of this attribute is 1, the DiskGroup online function starts all volumes belonging to that disk group after importing the group. **Note:** With VxVM version 5.1.100.0 onwards, if the Veritas Volume Manager default autostartvolumes at system level is set to on, all the volumes of the disk group will be started as a part of the import disk group. Type and dimension: boolean-scalar Default: 1

Table 2-2 Optional attributes

Optional attributes	Description
StopVolumes	If the value of this attribute is 1, the DiskGroup offline function stops all volumes belonging to that disk group before it deports the disk group. Type and dimension: boolean-scalar Default: 1
UmountVolumes	This attribute enables the DiskGroup resource to forcefully go offline even if open volumes are mounted outside of VCS control. When the value of this attribute is 1 and the disk group has open volumes, the following occurs: ■ The agent attempts to unmount the file systems on open volumes. If required, the agent attempts to kill all VCS managed and un-managed applications using the file systems on those open volumes. ■ The agent attempts to forcefully unmount the file systems to close the volumes. Type and dimension: integer-scalar Default: 0
MonitorReservation	If the value of this attribute is 1, and SCSI-3 fencing is used, the agent monitors the SCSI reservation on the disk group. If the reservation is missing, the Monitor agent function takes the resource offline. Type and dimension: boolean-scalar Default: 0

Table 2-2 Optional attributes

Optional attributes	Description
PanicSystemOnDGLoss	Determines whether to panic the node if the disk group becomes disabled. A loss of storage connectivity can cause the disk group to become disabled. If the value of this attribute is 1, and the disk group becomes disabled, the node panics. If the value of this attribute is 1, and the Monitor agent function (entry point) hangs a consecutive number of times per the value of the FaultOnMonitorTimeouts attribute, then the node panics. **Note:** System administrators may want to set a high value for FaultOnMonitorTimeout to increase system tolerance. If the value of the attribute is 0, and the disk group becomes disabled, the following occurs: ■ If the cluster has I/O fencing enabled, the DiskGroup resource is marked FAULTED. This state results in the agent attempting to take the service group offline. As part of bringing the DiskGroup resource offline, the agent attempts to deport the disabled disk group. Even if disabled disk group fails to deport, the DiskGroup resource enters a FAULTED state. This state enables the failover of the service group that contains the resource. To fail back the DiskGroup resource, manually deport the disk group after restoring storage connectivity. ■ If the cluster does not use I/O fencing, a message is logged and the resource is reported ONLINE. The resource is reported ONLINE so that it does not fail over, which ensures data integrity. **Note:** The PanicSystemOnDGLoss attribute does not depend on the MonitorReservation attribute. Type and dimension: boolean-scalar Default: 0

Table 2-2 Optional attributes

Optional attributes	Description
Reservation	Determines if you want to enable SCSI-3 reservation. This attribute can have one of the following three values: ■ ClusterDefault—The disk group is imported with SCSI-3 reservation if the value of the cluster-level UseFence attribute is SCSI3. If the value of the cluster-level UseFence attribute is NONE, the disk group is imported without reservation. ■ SCSI3—The disk group is imported with SCSI-3 reservation if the value of the cluster-level UseFence attribute is SCSI3. ■ NONE—The disk group is imported without SCSI-3 reservation. Type and dimension: string-scalar Default: ClusterDefault Example: "SCSI3"

Table 2-3 Internal attributes

Required attribute	Description
tempUseFence	Do not use. For internal use only.
NumThreads	Number of threads used within the agent process for managing resources. This number does not include the threads used for internal purposes. Do not modify this attribute for this agent. Setting this attribute to a higher value may result in agent function timeouts due to serialization of underlying commands. Type and dimension: static integer-scalar Default: 1

Resource type definition

```
type DiskGroup (
    static keylist SupportedActions = { "license.vfd", "disk.vfd",
    "udid.vfd", "verifyplex.vfd", checkudid, numdisks, campusplex,
    volinuse, joindg, splitdg, getvxvminfo }
    static int OnlineRetryLimit = 1
    static str ArgList[] = { DiskGroup, StartVolumes, StopVolumes,
    MonitorOnly, MonitorReservation, tempUseFence,
    PanicSystemOnDGLoss, UmountVolumes, Reservation }
    str DiskGroup
    boolean StartVolumes = 1
    boolean StopVolumes = 1
    static int NumThreads = 1
    boolean MonitorReservation = 0
    temp str tempUseFence = INVALID
    boolean PanicSystemOnDGLoss = 0
    int UmountVolumes = 0
    str Reservation = ClusterDefault
)
```

DiskGroup agent notes

The DiskGroup agent has the following notes:

High availability fire drill

The high availability fire drill detects discrepancies between the VCS configuration and the underlying infrastructure on a node. These discrepancies might prevent a service group from going online on a specific node.

For DiskGroup resources, the high availability fire drill checks for:

- The Veritas Volume Manager license
- Visibility from host for all disks in the disk group
- The same disks for the disk group on cluster nodes
- Equal number of plexes on all sites for the disk group in a campus cluster setup

For more information about using the high availability fire drill see the *Veritas Cluster Server Administrator's Guide*.

Using volume sets

When you use a volume set, set StartVolumes and StopVolumes attributes of the DiskGroup resource that contains a volume set to 1. If a file system is created on the volume set, use a Mount resource to mount the volume set.

See the Mount agent description for more information.

Setting the noautoimport flag for a disk group

VCS requires that the noautoimport flag of an imported disk group be explicitly set to true. This value enables VCS to control the importation and deportation of disk groups as needed when bringing disk groups online and taking them offline.

To check the status of the noautoimport flag for an imported disk group

- # **vxprint -l *disk_group* | grep noautoimport**
  If the output from this command is blank, the noautoimport flag is set to false and VCS lacks the necessary control.

For VxVM version 5.0 or later on AIX

The Monitor function changes the value of the VxVM noautoimport flag from off to on. It changes the value instead of taking the service group offline. This action allows VCS to maintain control of importing the disk group.

The following command changes the autoimport flag to false:

```
# vxdg -g disk_group set autoimport=no
```

For VxVM version 4.0

When you enable a disk group that is configured as a DiskGroup resource that does not have the noautoimport flag set to true, VCS forcibly deports the disk group. This forcible deportation may disrupt applications running on the disk group.

To explicitly set the noautoimport flag to true, deport the disk group and import it with the -t option as follows:

To deport the disk group, enter:

```
# vxdg deport disk_group
```

To import the disk group, specifying the noautoimport flag be set to true to ensure that the disk group is not automatically imported, enter:

```
# vxdg -t import disk_group
```

Configuring the Fiber Channel adapter

You must set FC adapter tunables appropriately to avoid excessive waits for monitor timeouts. One FS adapter tunable is FC error recovery policy.

Refer to the Fiber Channel adapter's configuration guide for further information.

Sample configurations

DiskGroup resource configuration

Example of a disk group resource in the Share Out mode.

```
DiskGroup dg1 (
    DiskGroup = testdg_1
)
```

Debug log levels

The DiskGroup agent uses the following debug log levels:

DBG_1, DBG_5

DiskGroupSnap agent

Use the DiskGroupSnap agent to perform fire drills in a campus cluster. The DiskGroupSnap agent enables you to verify the configuration and data integrity in a Campus Cluster environment with VxVM stretch mirroring. The agent also supports SCSI-3 fencing.

For more information on fire drills, refer to the *Veritas Cluster Server Administrator's Guide*.

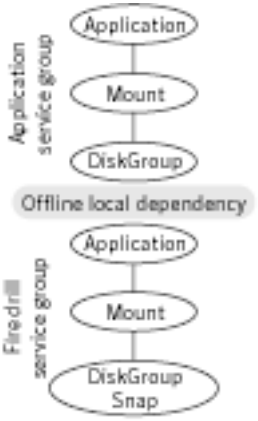
For important information about this agent, refer to:

"DiskGroupSnap agent notes" on page 38

Dependencies

The DiskGroupSnap resource does not depend on any other resources. The service group that contains the DiskGroupSnap agent's resource has an offline local dependency on the application's service group. The offline local dependency is to make sure the firedrill service group and the application service group are not online at the same site at the same time.

Figure 2-2 Sample service group that includes a DiskGroupSnap resource



Agent functions

Online	Verifies that the application's disk group is in a valid campus cluster configuration. It detaches the site that the value of the FDSiteName attribute specifies. It then creates another disk group to be used for the fire drill on the detached site.
Offline	This re-attaches the site that the value of the FDSiteName attribute specifies back to the application's disk group.
Monitor	Monitors the DiskGroupSnap resource.
Clean	Takes the DiskGroupSnap resource offline.
Open	If the DiskGroupSnap resource has a parent resource that is not ONLINE, then it deletes the online lock file of the DiskGroupSnap resource. This marks the DiskGroupSnap resource as OFFLINE.

State definitions

ONLINE	The DiskGroupSnap resource functions normally.
OFFLINE	The DiskGroupSnap resource is not running.
UNKNOWN	A configuration error exists.
FAULTED	The DiskGroupSnap resource is taken offline unexpectedly outside of VCS control.

Attributes

Table 2-4 Required attributes

Required attribute	Description
TargetResName	The name of the DiskGroup resource from the application service group. Type-dimension: string-scalar Example: "dgres"

FDSiteName	At a site, this is the unique VxVM site name tag for the fire drill disks. You can run the fire drill in the following configurations:
	■ In the Gold configuration, a site has a dedicated set of fire drill disks. In Figure 2-4, the disaster recovery site uses a Gold configuration.
	■ In the Bronze configuration, a site uses its data disks as fire drill disks. In Figure 2-4, the primary site uses a Bronze configuration.
	Type and dimension: string-scalar
	Example:
	The value for the FDSiteName attribute for the configuration for Figure 2-4 is:
	`"FDSiteName@Node_A = pri"`
	`"FDSiteName@Node_B = pri"`
	`"FDSiteName@Node_C = dr_fd"`
	`"FDSiteName@Node_D = dr_fd"`

Table 2-5 Internal attribute

Required attribute	Description
NumThreads	Number of threads used within the agent process for managing resources. This number does not include threads used for other internal purposes. Do not modify this attribute for this agent. Setting this attribute to a higher value may result in agent function timeouts due to serialization of underlying commands. Type and dimension: integer-scalar Default: 1

DiskGroupSnap agent notes

The DiskGroupSnap agent has the following notes:

- "Configuring the SystemZones attribute for the fire drill service group" on page 38
- "Configuring the firedrill service group" on page 39
- "Adding the ReuseMntPt attribute to the ArgList attribute for the Mount agent type" on page 39
- "Configuration considerations" on page 39
- "Agent limitations" on page 40

Configuring the SystemZones attribute for the fire drill service group

You must assign the local system values to the SystemZones attribute of the application's service group. You set these values so that the service group fails over in the same zone before it tries to fail over across zones. For more information about campus cluster setup, refer to the *Veritas Cluster Server Administrator's Guide*.

For example, you set up the service group's SystemZones attribute for two zones: 0 and 1. You want the service group on Node_A and Node_B to fail over between the two nodes before it comes up on Node_C and Node_D. The application and its fire drill service group both have the following values for the SystemZones attribute:

```
SystemZones = { Node_A = 0, Node_B = 0, Node_C = 1, Node_D = 1 }
```

Configuring the firedrill service group

In the firedrill service group, the application-level resources (for example, process resources, application resources, or Oracle resources, and so on) can have the same attribute values in the firedrill service group and the application service group. The reuse of the same values for the attributes can result in VCS reporting the wrong resources as online.

Set the FireDrill type-level attribute to 1 for those types. For example, if the Oracle and Listener resources are configured identically, set the FireDrill attribute for Oracle and Listener to 1:

```
haconf –makerw
hatype –modify Oracle FireDrill 1
hatype –modify Listener FireDrill 1
haconf -dump –makero
```

Adding the ReuseMntPt attribute to the ArgList attribute for the Mount agent type

If you plan to use a Mount resource in a firedrill service group, you must add the ReuseMntPt attribute to ArgList and set its value to 1.

To add the ReuseMntPt attribute to the ArgList attribute and set its value to 1

1. Make the configuration read and write.
   # **haconf -makerw**
2. Add the ReuseMntPt attribute to the ArgList attribute.
   # **hatype -modify Mount ArgList -add ReuseMntPt**
3. Change the value of the ReuseMntPt attribute to 1 for the firedrill's Mount resource.
   # **hares -modify *firedrill_mount_resource_name* ReuseMntPt 1**
4. Change the value of the ReuseMntPt attribute to 1 for the original Mount resource.
   # **hares -modify *original_mount_resource_name* ReuseMntPt 1**
5. Make the configuration read only.
   # **haconf –dump –makero**

Configuration considerations

Keep the following recommendations in mind:

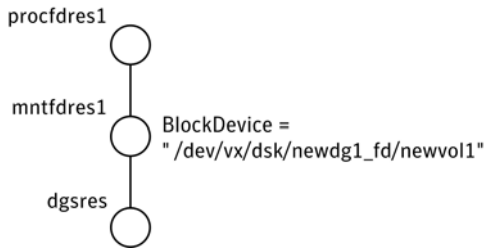
- You must install Veritas Volume Manager 5.1 or later with the FMR license and the Site Awareness license.
- Do not bring the DiskGroupSnap resource online in the SystemZone where the application service group is online.

- Make sure that the firedrill service group and the application service group both use the same values for the SystemZones attribute.
- Do not use Volume resources in the firedrill service group. The DiskGroupSnap agent internally uses the `vxvol` command to start all the volumes in the firedrill disk group.
- In large setups, you may need to tweak the various timer values so that the timers do not time out while waiting for VxVM commands to complete. The timers you need to tweak are the OfflineTimeout for the DiskGroupSnap resource and MonitorInterval and ActionTimeout for the associated DiskGroup resource, for example:

```
haconf -makerw
hares -override dgsres OfflineTimeout
hares -modify dgsres OfflineTimeout 600
hares -override dgres MonitorInterval
hares -modify dgres MonitorInterval 1200 (this has to be twice
the value intended for ActionTimeout below)
hares -override dgres ActionTimeout
hares -modify dgres ActionTimeout 600
haconf -dump -makero
```

- When you create the firedrill service group, in general use the same attribute values that you use in the application service group.
  The BlockDevice attribute of the Mount resource changes between the application service group and the firedrill service group. In the BlockDevice path, you must append an _fd to the disk group name portion, for example, /dev/vx/dsk/newdg1/newvol1 becomes /dev/vx/dsk/newdg1_fd/newvol1.
  Figure 2-3 shows the changes to resource values for the firedrill service group; note that the Volume resource is not included.

Figure 2-3 Sample resource values for a DiskGroupSnap resource



Agent limitations

The following limitations apply to the DiskGroupSnap agent:

- The DiskGroupSnap agent does not support Volume Sets.

- The DiskGroupSnap agent cannot be used in a Storage Foundation RAC environment.
- The online and offline operations of the DiskGroupSnap resource invokes VCS action entry points to run VxVM commands to detach/reattach the fire drill site. Since VxVM requires that these commands are run on the node where the disk group is imported, the disk group has to be imported on some node in the cluster before these operations.
- Take the firedrill service group offline before you shut down VCS on any node. If you fail to take the firedrill service group offline before you shut down VCS, you must manually reattach the fire drill site to the disk group to continue to perform fire drills.
- Use the enclosures that have the ASL/APM libraries that are supported in the Veritas Volume Manager. To view the supported enclosures, use the `vxddladm listsupport` command.

Resource type definition

```
type DiskGroupSnap (
    static int ActionTimeout = 120
    static int MonitorInterval = 300
    static int NumThreads = 1
    static str ArgList[] = { TargetResName, FDSiteName }
    str TargetResName
    str FDSiteName
)
```

Sample configurations

In Figure 2-4, the Primary site is in the Bronze configuration and the Disaster recovery site is in a Gold configuration.

Since the Primary site does not have dedicated fire drill disks, it is in a Bronze configuration. In the Bronze configuration, you re-purpose the mirror disks in the disaster recovery site to serve as fire drill test disks. The drawback with the Bronze configuration is that if a disk failure occurs when the fire drill is online at the Primary site, it results in a site failure.

The FDSiteName value in a bronze configuration is the VxVM site name. For this configuration, the FDSiteName attribute values for the nodes at the Primary site follow:

```
FDSiteName@Node_A = pri
FDSiteName@Node_B = pri
```

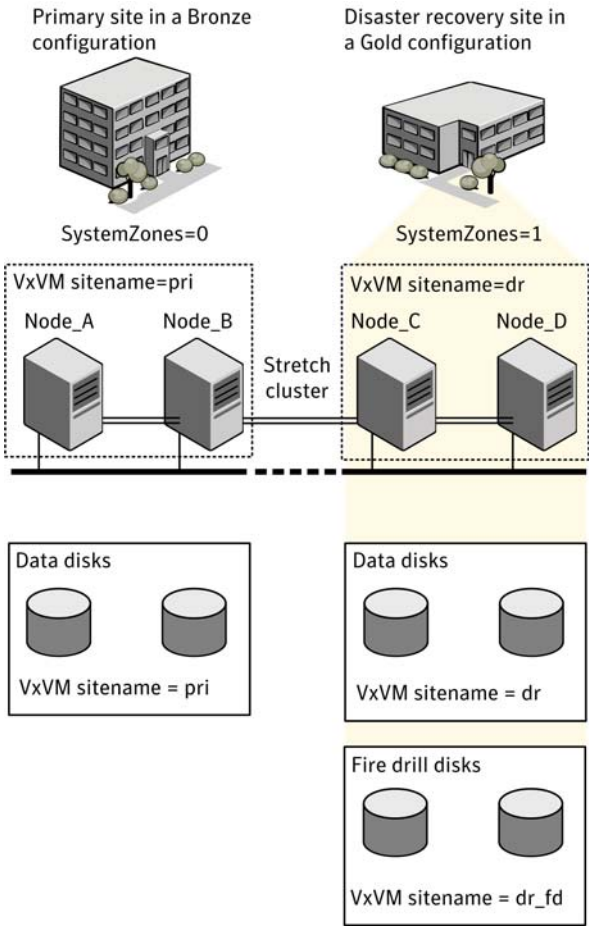
The Disaster Recovery site is in a Gold configuration as it has dedicated fire drill disks at the site. For the FDSiteName attribute, use the VxVM site tag given to the fire drill disks. For this configuration, the FDSiteName attribute values for the nodes at the Disaster recovery site follow:

```
FDSiteName@Node_C = dr_fd
FDSiteName@Node_D = dr_fd
```

Set values for the SystemZones attribute to zero for Node_A and Node_B, and one for Node_C and Node_D. For example:

```
SystemZones = { Node_A = 0, Node_B = 0, Node_C = 1, Node_D = 1 }
```

Figure 2-4 Primary site with the Bronze configuration and the disaster recovery site with the Gold configuration



Typical main.cf configuration

The following sample configuration shows the fire drill's service group and its corresponding application service group. The fire drill's service group follows:

```
group dgfdsg (
    SystemList = { Node_A = 0, Node_B = 1, Node_C = 2, Node_D = 3 }
    SystemZones = { Node_A = 0, Node_B = 0, Node_C = 1, Node_D = 1}
    )

    DiskGroupSnap dgsres (
        TargetResName = dgres
```

```
        FDSiteName @Node_A = pri
        FDSiteName @Node_B = pri
        FDSiteName @Node_C = dr_fd
        FDSiteName @Node_D = dr_fd
        )

    Mount mntfdres1 (
        MountPoint = "/dgsfs1"
        BlockDevice = "/dev/vx/dsk/newdg1_fd/newvol1"
        FSType = vxfs
        FsckOpt = "-y"
        ReuseMntPt  = 1
        )

    Mount mntfdres2 (
        MountPoint = "/dgsfs2"
        BlockDevice = "/dev/vx/dsk/newdg1_fd/newvol2"
        FSType = vxfs
        FsckOpt = "-y"
        ReuseMntPt  = 1
        )

    Process procfdres1 (
        PathName = "/usr/bin/ksh"
        Arguments = "/scrib.sh /dgsfs1"
        )

    Process procfdres2 (
        PathName = "/usr/bin/ksh"
        Arguments = "/scrib.sh /dgsfs2"
        )

    requires group dgsg offline local
    mntfdres1 requires dgsres
    mntfdres2 requires dgsres
    procfdres1 requires mntfdres1
    procfdres2 requires mntfdres2
```

The application's service group (the actual service group) follows:

```
group dgsg (
    SystemList = { Node_A = 0, Node_B = 1, Node_C = 2, Node_D = 3 }
    SystemZones = { Node_A = 0, Node_B = 0, Node_C = 1, Node_D = 1}
    )

    DiskGroup dgres (
        DiskGroup = newdg1
        )

    Mount mntres1 (
        MountPoint = "/dgsfs1"
        BlockDevice = "/dev/vx/dsk/newdg1/newvol1"
        FSType = vxfs
```

```
        FsckOpt = "-y"
        ReuseMntPt  = 1
        )

    Mount mntres2 (
        MountPoint = "/dgsfs2"
        BlockDevice = "/dev/vx/dsk/newdg1/newvol2"
        FSType = vxfs
        FsckOpt = "-y"
        ReuseMntPt  = 1
        )

    Process procres1 (
        PathName = "/usr/bin/ksh"
        Arguments = "/scrib.sh /dgsfs1"
        )

    Process procres2 (
        PathName = "/usr/bin/ksh"
        Arguments = "/scrib.sh /dgsfs2"
        )

    mntres1 requires dgres
    mntres2 requires dgres
    procres1 requires mntres1
    procres2 requires mntres2
```

Oracle main.cf configuration

The following Oracle configuration has been simplified for presentation within this guide. Note that *NIC0* represents the NIC's name.

```
group fd_oragrp (
        SystemList = { Node_A = 0, Node_B = 1 }
        AutoStart = 0
        SystemZones = { Node_A = 0, Node_B = 1 }
        )

        DiskGroupSnap dgres (
                FDSiteName @Node_A = siteA
                FDSiteName @Node_B = siteB
                TargetResName = oradg_res
                )

        IP fd_oraip (
                Device = NIC0
                Address = "10.198.95.191"
                )

        Mount fd_archmnt (
                FsckOpt = "-y"
```

```
                ReuseMntPt = 1
                BlockDevice = "/dev/vx/dsk/oradg_fd/archive_vol"
                MountPoint = "/ora_archive"
                FSType = vxfs
                )

        Mount fd_datamnt (
                FsckOpt = "-y"
                ReuseMntPt = 1
                BlockDevice = "/dev/vx/dsk/oradg_fd/data_vol"
                MountPoint = "/ora_data"
                FSType = vxfs
                )

        NIC fd_oranic (
                Device = NIC0
                )

        Netlsnr fd_LSNR (
                Home = "/opt/oracle/ora_home"
                Owner = oracle
                )

        Oracle fd_Ora_01 (
                Owner = oracle
                Home = "/opt/oracle/ora_home"
                Sid = Ora_01
                )

    requires group oragrp offline local
    fd_LSNR requires fd_Ora_01
    fd_LSNR requires fd_oraip
    fd_Ora_01 requires fd_archmnt
    fd_Ora_01 requires fd_datamnt
    fd_archmnt requires dgres
    fd_datamnt requires dgres
    fd_oraip requires fd_oranic

group oragrp (
        SystemList = { Node_A = 0, Node_B = 1 }
        AutoStartList = { Node_A, Node_B }
        SystemZones = { Node_A = 0, Node_B = 1 }
        )

        DiskGroup oradg_res (
                DiskGroup = oradg
                )

        IP Node_A4vip (
                Device = NIC0
                Address = "10.198.95.192"
                )
```

```
    Mount arch_mnt (
            FsckOpt = "-y"
            ReuseMntPt = 1
            BlockDevice = "/dev/vx/dsk/oradg/archive_vol"
            MountPoint = "/ora_archive"
            FSType = vxfs
            )

    Mount data_mnt (
            FsckOpt = "-y"
            ReuseMntPt = 1
            BlockDevice = "/dev/vx/dsk/oradg/data_vol"
            MountPoint = "/ora_data"
            FSType = vxfs
            )

    NIC nic_Node_A4vip (
            Device = NIC0
            )

    Netlsnr LSNR (
            Home = "/opt/oracle/ora_home"
            Owner = oracle
            )

    Oracle Ora_01 (
            Owner = oracle
            Home = "/opt/oracle/ora_home"
            Sid = Ora_01
            )

    Volume arch_vol (
            Volume = archive_vol
            DiskGroup = oradg
            )

    Volume data_vol (
            Volume = data_vol
            DiskGroup = oradg
            )

LSNR requires Ora_01
LSNR requires Node_A4vip
Ora_01 requires arch_mnt
Ora_01 requires data_mnt
arch_mnt requires arch_vol
arch_vol requires oradg_res
data_mnt requires data_vol
data_vol requires oradg_res
Node_A4vip requires nic_Node_A4vip
```

Debug log levels

The DiskGroupSnap agent uses the following debug log levels:

DBG_1

Volume agent

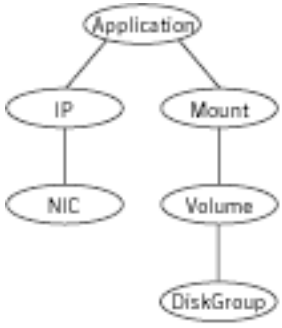
The Volume agent brings online, takes offline, and monitors a Veritas Volume Manager (VxVM) volume. Use the agent to make a volume highly available.

Note: Do not use the Volume agent for volumes created for replication.

Dependencies

Volume resources depend on DiskGroup resources.

Figure 2-5 Sample service group that includes a Volume resource



Agent functions

Online	Uses the `vxrecover` command to start the volume.
Offline	Uses the `vxvol` command to stop the volume.
Monitor	Attempts to read a block from the raw device interface to the volume to determine if the volume is online, offline, or unknown.
Clean	Terminates all ongoing resource actions and takes the resource offline—forcibly when necessary.

State definitions

ONLINE	Indicates that the specified volume is started and that I/O is permitted.

OFFLINE	Indicates that the specified volume is not started and that I/O is not permitted.
FAULTED	Indicates the volume stopped unexpectedly and that I/O is not permitted.
UNKNOWN	Indicates that the agent could not determine the state of the resource or that the resource attributes are configured incorrectly.

Attributes

Table 2-6 Required attributes

Required attribute	Description
DiskGroup	Name of the disk group that contains the volume. Type and dimension: string-scalar Example: `"DG1"`
Volume	Name of the volume from disk group specified in DiskGroup attribute. Type and dimension: string-scalar Example: "DG1Vol1"

Table 2-7 Internal attribute

Internal attribute	Description
NumThreads	Number of threads used within the agent process for managing resources. This number does not include threads used for other internal purposes. Do not modify this attribute for this agent. Setting this attribute to a higher value may result in agent function timeouts due to serialization of underlying commands. Default: 1

Resource type definition

```
type Volume (
    static int NumThreads = 1
    static str ArgList[] = { Volume, DiskGroup }
    str Volume
    str DiskGroup
)
```

Sample configuration

```
Volume sharedg_vol3 (
    Volume = vol3
    DiskGroup = sharedg
)
```

Debug log levels

The Volume agent uses the following debug log levels:

DBG_1, DBG_3, DBG_5

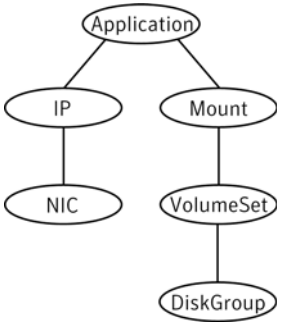
VolumeSet agent

The VolumeSet agent brings online, takes offline, and monitors a Veritas Volume Manager (VxVM) volume set. Use the agent to make a volume set highly available.

Dependencies

VolumeSet resources depend on DiskGroup resources.

Figure 2-6 Sample service group that includes a VolumeSet resource



Agent functions

Online	Uses the `vxrecover` command to start the volume set.
Offline	Uses the `vxvol` command to stop the volume set.
Monitor	Attempts to read a block from the raw device interface to the volumes inside the volume set to determine if the volume set is online, offline, or unknown.
Clean	Terminates all ongoing resource actions and takes the resource offline—forcibly when necessary.

State definitions

ONLINE	Indicates that all the volumes in the volume set are started and that I/O is permitted for all the volumes.

OFFLINE	Indicates that at least one of the volume is not started in the volume set and that I/O is not permitted for that volume.
FAULTED	Indicates the volumes that are inside the volume set have stopped unexpectedly and that I/O is not permitted.
UNKNOWN	Indicates that the agent could not determine the state of the resource or that the resource attributes are configured incorrectly.

Attributes

Table 2-8 Required attributes

Required attribute	Description
DiskGroup	The name of the disk group that contains the volume set. Type and dimension: string-scalar Example: "DG1"
VolumeSet	The name of the volume set from the disk group that you specified in the DiskGroup attribute. Type and dimension: string-scalar Example: "DG1VolSet1"

Resource type definition

```
type VolumeSet (
    static str ArgList[] = { DiskGroup, VolumeSet }
    str VolumeSet
    str DiskGroup
    )
```

Sample configurations

This sections contains sample configurations for this agent.

A configured VolumeSet that is dependent on a DiskGroup resource

The VolumeSet's shared_vset3 resource is configured and is dependent on DiskGroup resource with a shared diskgroup.

```
VolumeSet sharedg_vset3 (
    VolumeSet = vset3
    DiskGroup = sharedg
)
```

Agent notes

This sections contains notes about this agent.

Inaccessible volumes prevent the VolumeSet agent from coming online

The VolumeSet agent does not come online if any volume is inaccessible in its volume set.

To remove a volume from volume set

- Enter the following commands to remove a volume from a volume set mounted on *mountpoint*.

  ```
  # fsvoladm remove mountpoint volume_name
  # vxvset -g diskgroup rmvol volumeset volume_name
  ```

Debug log levels

The VolumeSet agent uses the following debug log levels:

DBG_1, DBG_4

LVMVG agent

The LVMVG agent activates, deactivates, and monitors a Logical Volume Manager (LVM) volume group. The LVMVG agent supports JFS or JFS2. It does not support VxFS. This agent ensures that the ODM (Object Data Manager) is in sync with changes to the volume group. Specifically from the last time that the volume group was imported on the system.

The LVMVG agent is also capable of ensuring high availability for AIX scalable volume group.

This agent supports Veritas Dynamic Multi-Pathing.

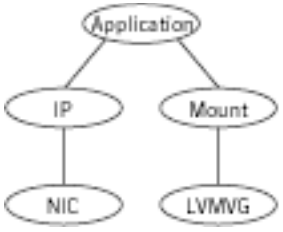
For important information on this agent, refer to:

"LVMVG agent notes" on page 59

Dependencies

No dependencies exist for the LVMVG resource.

Figure 2-7 Sample service group for an LVMVG resource



Agent functions

Online	Activates the volume group. The Online agent function expects that the volume group is already imported on the system. If the volume group had been modified on a system where it was previously active, the online agent function detects the modification. It then syncs up the ODM on the system where you want to bring the volume group resource online.
Offline	Deactivates the volume group.
Monitor	Determines the volume group's state (activated or deactivated) and availability for read/write operations.

Clean	Terminates all ongoing resource actions and takes the resource offline, forcibly when necessary.
Action	Different action agent functions follow: ■ pv.vfd Checks if all the disks in the volume group are visible on a host. If it fails, check if the path to disks exists from the host and check if LUN masking and zoning are set properly. ■ autoon.vfd Checks if the flag to automatically activate volume group on system restart is set to yes. If it fails, set the “auto on” flag of volume group to “no”. ■ volinuse Checks if open volumes are in use or file systems on volumes that are mounted outside of VCS configuration.

State definitions

ONLINE	Indicates that the volume group is activated.
OFFLINE	Indicates that the volume group is deactivated.
FAULTED	Indicates that the volume group has unexpectedly deactivated or deported or been disabled.
UNKNOWN	Indicates that a problem exists either with the configuration or the ability to determine the status of the resource.

Attributes

Table 2-9 Required attributes

Required attribute	Description
MajorNumber	Integer that represents the major number of the volume group. To ensure NFS functions properly, assign the same major number to the volume group on each system in the cluster. Type and dimension: integer-scalar

Table 2-9 Required attributes

Required attribute	Description
NumThreads	The number of threads that are used within the agent process for managing resources. This number does not include the threads that are used for other internal purposes. This resource type attribute is for internal use only. This value of this attribute must be set to 1. Type and dimension: integer-scalar Default: 1
VolumeGroup	Name of the volume group that is configured with LVM. Type and dimension: string-scalar Example: "testvg1"

Table 2-10 Optional attributes

Optional attribute	Description
GroupName	Attribute used to specify the volume's group. If set, the groups's name is applied to the volume group and all of its logical volumes. Type and dimension: string-scalar Default: system
ImportvgOpt	Attribute used to specify options for the importvg command. The default option, "n", indicates the volume group is not automatically activated when imported. Type and dimension: string-scalar Default: n

Table 2-10 Optional attributes

Optional attribute	Description
Mode	Attribute used to specify permissions for a volume group and its logical volumes. If set, these permissions are applied to the volume group and all of its logical volumes. Type and dimension: string-scalar Default: 640
OwnerName	Attribute used to specify the volume owner's name. If set, the owner's name is applied to the volume group and all of its logical volumes. Type and dimension: string-scalar Default: root
SyncODM	Integer that specifies whether or not the agent ensures that the ODM is in sync with any changes to the volume group. If the value of this attribute is 1, the agent ensures that the ODM is in sync with the changes to the volume group. In situations where the volume group was modified on another system in the cluster. The sync operation occurs on the system where the agent brings the volume group online. If the value of this attribute is 0, the changes to the volume group are independent of the ODM. See "SyncODM Attribute" on page 62. Type and dimension: integer-scalar Default: 1
VaryonvgOpt	Attribute used to specify options for the varyonvg command. Type and dimension: string-scalar

Table 2-10 Optional attributes

Optional attribute	Description
ModePermSyncFlag	Integer that specifies whether or not the agent attempts to varyonvg before getting the mode and permissions to preserves during ODM sync or not. If the value of this attribute is 0, the agent will get the mode and permissions without varyonvg. The mode and permissions will be applied after the sync is complete. If the value of this attribute is 1 (default), the agent will varyonvg and then get the mode and permissions to be reapplied after sync is complete. Type and dimension: integer_scalar Default: 1

Resource type definition

```
type LVMVG (
    static keylist SupportedActions = { "pv.vfd", numdisks,
    "autoon.vfd", volinuse }
    static int NumThreads = 1
    static str ArgList[] = { VolumeGroup, MajorNumber, OwnerName,
    GroupName,  Mode, ImportvgOpt, VaryonvgOpt, SyncODM,
    ModePermSyncFlag }
    str VolumeGroup
    int MajorNumber
    str OwnerName
    str GroupName
    str Mode
    str ImportvgOpt = n
    str VaryonvgOpt
    int SyncODM = 1
    int ModePermSyncFlag = 1
)
```

LVMVG agent notes

The LVMVG agent for AIX has the following notes:

- "LVMVG support in a VIO server environment" on page 60
- "Deactivation failure using the varyoffvg command on losing storage connectivity" on page 60
- "LVMVG Agent Supports JFS or JFS2" on page 61

LVMVG support in a VIO server environment

The LVMVG agent supports volume groups created with virtual SCSI devices. AIX and VIOS must be at the following required levels:

- The AIX operating system level must be AIX 5.3 TL7 SP6 or later and AIX 6.1 TL2 or later.

  For more information about supported AIX versions, refer to the *Veritas Cluster Server Installation Guide*.
- The VIOS version must be VIOS 1.3 Fix Pack 8.1 or later.

Deactivation failure using the varyoffvg command on losing storage connectivity

In certain circumstances, the varyoffvg command does not deactivate all the volume groups on a node. This failure can prevent the failback of the LVMVG resource.

In situations where storage connectivity is lost, the LVMVG resources fails over. Failback for the LVMVG resource requires the deactivation of the volume groups on the node that lost its connectivity to storage. VCS uses the varyoffvg command to deactivate the volume groups. The LVMVG resource cannot fail back, however, when deactivation is unsuccessful.

When the volume group loses its storage connectivity, the clean function executes the `varyoffvg` command. Deactivation using the `varyoffvg` command can fail, however, if the volume group is busy. Criteria that can cause this failure can include:

- when the volume group has pending I/O operations, or

- when an application or upper-level resources in the resource dependency tree uses the volume group.

To overcome this deactivation failure, a post offline trigger has been added to issue the `varyoffvg` command. A side effect of the post offline trigger is that you must set the value of the OnlineRetryLimit attribute to 0.

After the restoration of storage connectivity, you must ensure that the volume groups are deactivated on the node. You can then clear the fault on the resources. If you find active volume groups, deactivate them using the varyoffvg command.

The LVMVG resource must be the bottom-most resource in the resource dependency tree in the service group. A resource under the LVMVG resource can potentially fail to go offline if the volume group's deactivation fails.

LVMVG Agent Supports JFS or JFS2

The LVMVG agent supports these file systems: JFS or JFS2. It does not support VxFS.

Volume group needs to be imported

The LVMVG agent relies on the ODM to find out the names of the disk devices that a volume group is created on. Unless a volume group is imported on the system, the ODM on that system does not contain any information about that volume group. Therefore, you must import the volume group on all the systems in the group's SystemList for the LVMVG agent to function properly.

For example, the volume groups (vg1 and vg2) must be imported on the specified systems (sysA and sysB).

See "LVMVG agent notes" on page 59.

Varyonvg options

By default, the agent checks the state of the disk devices underneath the volume group. If the disk device is in a defined state, the agent resets it to an available state. You can use the VaryonvgOpt attribute to change this default behavior.

You can tell the agent not to check for the state of the disk devices. Set the VaryonvgOpt attribute in the main.cf file to a value of "u". This option to the `varyonvg` command ensures that the disks underneath the volume group are not reserved when the volume group is activated.

Note: When you activate a volume group with the "u" option, ghost disks are not created. Therefore, you do not have to reset disks for these volume groups.

SyncODM Attribute

The LVMVG agent ensures that the ODM is in sync with any changes to the volume group since it was last imported on the system. This sync happens only if this attribute is set to 1. The agent maintains a time stamp file, /var/VRTSvcs/log/tmp/*volume_group_name*.ts, which records the time when the volume group was last imported on the system. When the agent initially brings a volume group online, the agent exports and reimports the group while initializing the time stamp file for that group. During the export and re-import processes, the agent preserves the ownership and mode information for the volume group and all its logical volumes.

The sync operation occurs when the time stamp value in the volume group's time stamp file is older than the time stamp value in the volume group's descriptor area. The timestamp value in the VGDA area of a volume group is updated after creating or deleting logical volumes, and adding or removing physical volumes.

Major Numbers

If a file system on a volume group is shared for NFS, make sure that the volume group is imported with the same major number. The volume group is imported on all of the nodes in the cluster.

To view a list of available major numbers on the system, enter the lvlstmajor command. For example:

```
# lvlstmajor
  49, 60 ...
```

To import volume group vg00 with major number 60, enter:

```
# importvg -V 60 -y vg00 hdisk3
```

To view the major number that is assigned to a volume group, use the `ls` command with the -l option. For example:

```
# ls -l /dev/vg00
crw-r-----   1 root     system    60,  0 Apr  2 16:05 /dev/vg00
```

Assign the same major number to the volume group on each system in the cluster. Specify this major number in the MajorNumber attribute of the LVMVG configuration.

Note: Do not specify the V option in the ImportvgOpt attribute string, the agent specifies this option.

Autoactivate Options

The "Concurrent Capable" options for the `importvg` and `mkvg` commands that are used with HACMP are not required for VCS. If an LVM volume group is

placed under VCS control, the autoactivate options should be turned off. Do this using SMIT or through the command line.

From SMIT, set the following field values when creating or altering the volume group:

```
Activate volume group AUTOMATICALLY                 no
    at system restart?
Create VG Concurrent Capable?                       no
Auto-varyon in Concurrent Mode?                     no
```

From the command line, to view the current value for these fields, use the `lsattr` command.

For example:

```
# lsattr -El vg00
    vgserial_id  0001632f00004c00000000ee092b3bd8 N/A False
    auto_on      y                                N/A True
    conc_capable n                                N/A True
    conc_auto_on n                                N/A True
    timestamp    3ceff3390a8b1379                 N/A True
```

From the command line, to change the value for these fields, use the `chvg` command.

To change the value of `auto_on` to n:

1. Activate the volume group vg00 (if the volume group is not already activated):

   ```
   # varyonvg vg00
   ```

2. Run the `chvg` command:

   ```
   # chvg -a 'n' vg00
   ```

3. Verify the changes:

   ```
   # lsattr -El vg00
       vgserial_id  0001632f00004c00000000ee092b3bd8 N/A False
       auto_on      n                                N/A True
       conc_capable n                                N/A True
       conc_auto_on n                                N/A True
       timestamp    3ceff3390a8b1379                 N/A True
   ```

LVMVG agent support for the Subsystem Device Driver (SDD)

The LVMVG agent supports the IBM Multipathing SDD version 1.4.0.0 and later. If disks are under SDD control, create a volume group with vpath devices. Refer to the SDD Documentation for configuration and migration of volume groups.

SDD support requires the /usr/sbin/lquerypr command, which provides a set of persistent reserve functions. The lquerypr command tool comes with the SDD installation package.

LVMVG agent support for the Hitachi's HiCommand Dynamic Link Manager (HDLM)

The LVMVG agent supports the Hitachi's HiCommand Dynamic Link Manager. For the details of the array and HDLM versions supported, refer to the HCL.

Note that if disks are under HDLM control, create a volume group with HDLM devices (dlmfdrv*n*). Refer to the HDLM documentation for configuration and migration of volume groups.

LVMVG agent support for the EMC PowerPath

The LVMVG agent supports the EMC PowerPath. For the details of the array and PowerPath versions supported, refer to the HCL.

Note that if disks are under PowerPath control, create a volume group with PowerPath devices (hdiskpower*n*). Refer to the EMC PowerPath documentation for configuration and migration of volume groups.

The hadevice utility

The LVMVG agent provides the hadevice utility. This utility checks the status of a disk device and resets a disk device to an available state. The utility then breaks any SCSI reservations on a disk device. Its syntax is:

```
hadevice -c | -r | -b -p device_name
```

The five possible states of a disk device are: AVAILABLE, DEFINED AND RESERVED, DEFINED AND UNRESERVED, PERSISTENT RESERVATION, and AVAILABLE AND OPEN.

To check the state of a disk device, enter:

```
# hadevice -c device_name
```

The following commands locate and remove ghost disks for a disk device and break any SCSI reservation on the disk device. When the -p flag follows the -b flag, it breaks any previous SCSI reservation on the device. It then obtains and retains a new reservation on the device. For SDD (vpath) disks, ghost disks are not created. Both the -b and -r flags remove any persistent reservation and clear all reservation key registration on the device. The -p flag (retain reservation) is not applicable for SDD disks.

To break any SCSI reservations on the disk device, enter:

```
# hadevice -b device_name
```

To break any SCSI reservations on the disk device, and obtain and retain a new reservation on the device, enter:

```
# hadevice -b -p device_name
```

To locate and remove ghost disks, reset a disk device that is in a DEFINED state and put it into an AVAILABLE state, enter:

```
# hadevice -r device_name
```

Removing a ghost disk from VxVM control

If VxVM 5.0 is installed, you may need to remove a ghost disk from VxVM control before using hadevice utility (except -r option).

If you check the ghost disk's status using the `hadevice -c hdisk#` command, you get an error. The error reads: `V-16-10011-10237 Error opening the device /dev/hdisk# (The file access permissions do not allow the specified action.)` Check if the ghost disk is under VxVM control. You can do this using the `vxdisk -eq list` command. If the disk is under VxVM control, remove it using the `vxdisk rm vxvm_disk_name`.

In this example, hdisk4 is a ghost disk.

```
sysA# vxdisk -eq list
Disk_0          auto      -          -         LVM        disk0
HDS9500-ALUA0_0 auto      -          -         error      hdisk4
HDS9500-ALUA0_1 auto      -          -         online     hdisk2
HDS9500-ALUA0_2 auto      -          -         online     hdisk3

sysA# vxdisk rm HDS9500-ALUA0_0
```

Sample configuration

```
system sysA (
    )

system sysB (
    )

system sysC (
    )

group  lvmgroup (
    SystemList = { sysA = 0, sysB = 1 }
    AutoStartList = { sysA }
    )

LVMVG lvmvg_vg1 (
    VolumeGroup = vg1
    MajorNumber = 50
    )

LVMVG lvmvg_vg2 (
    VolumeGroup = vg2
    MajorNumber = 51
    ImportvgOpt = "f"
    )
```

Debug log levels

The LVMVG agent uses the following debug log levels:

DBG_1, DBG_2, DBG_3, DBG_5

Mount agent

The Mount agent brings online, takes offline, and monitors a file system or an NFS client mount point. You can use the agent to make file systems or NFS client mount points highly available. This agent is WPAR-aware. The ContainerOpts resource type attribute for this type has a default value of 0 for RunInContainer and a default value of 0 for PassCInfo. Symantec recommends that you do not change these values. Refer to the *Veritas Cluster Server Administrator's Guide* for more information.

This agent is IMF-aware and uses asynchronous monitoring framework (AMF) kernel driver for IMF notification. For more information about the Intelligent Monitoring Framework (IMF) and intelligent resource monitoring, refer to the *Veritas Cluster Server Administrator's Guide*.

Note: Intelligent Monitoring Framework for mounts is supported only for the following mount types: VxFS, CFS, and NFS.

The Mount agent supports the IPv6 protocol.

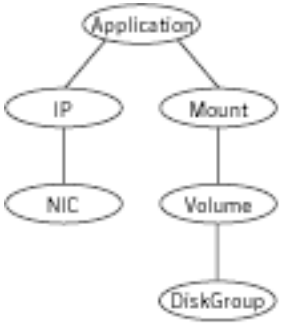
For important information about this agent, refer to:

"Mount agent notes" on page 78

Dependencies

The Mount resource does not depend on any other resources.

Figure 2-8 Sample service group that includes a Mount resource



Agent functions

Online	Mounts a block device on the directory. If the mount process fails for non-NFS mounts, the agent attempts to run the `fsck` command on the device before attempting to mount the file system again. If file system type is NFS, agent mounts the remote file system to a specified directory. The remote NFS file system is specified in the BlockDevice attribute.
Offline	Unmounts the mounted file system gracefully.
Monitor	Determines if the file system is mounted. If IMF is enabled for the Mount agent, the resource is monitored asynchronously and any change in the resource state is immediately sent to VCS for appropriate action.
imf_init	Initializes the agent to interface with the asynchronous monitoring framework (AMF) kernel driver. This function runs when the agent starts up.
imf_getnotification	Waits for notification about resource state changes. This function runs after the agent initializes with the AMF kernel driver. The agent continuously waits for notification and takes action on the resource upon notification.
imf_register	Registers the resource entities, which the agent must monitor, with the AMF kernel driver. This function runs for each resource after the resource goes into steady state (online or offline).
Clean	Unmounts the mounted file system forcefully.

Info

The Mount info agent function executes the command:

```
df -k mount_point
```

The output displays Mount resource information:

```
Size Used Avail Use%
```

To initiate the info agent function, set the InfoInterval timing to a value greater than 0. In this example, the info agent function executes every 60 seconds:

```
haconf -makerw
hatype -modify Mount InfoInterval 60
```

The command to retrieve information about the Mount resource is:

```
hares -value mountres ResourceInfo
```

Output includes:

```
Size 2097152
Used 139484
Available 1835332
Used% 8%
```

Action

- chgmntlock
  Resets the VxFS file system lock to a VCS-defined lock.
- mountpoint.vfd
  Checks if the specified mount point exists on the offline node. If it fails and you request that VCS fixes it, it creates the mount point directory using `mkdir` command.
- mounted.vfd
  Checks if the mount point is already mounted on the offline node. If it fails, you need to unmount all the file systems from the specified mount point directory.
- vxfslic.vfd
  Checks for valid Veritas File System (VxFS) licenses. If it fails, you need to update the license for VxFS.
- mountentry.vfd
  Checks that the mount point is not listed in auto file system tables (for example, /etc/filesystems).
  If this action fails, you need to remove the mount point from auto file system tables.

State definitions

ONLINE

For the local file system, indicates that the block device is mounted on the specified mount point.

For an NFS client, indicates that the NFS remote client is mounted on the specified mount directory.

OFFLINE	For the local file system, indicates that the block device is not mounted on the specified mount point. For an NFS client, indicates that the NFS remote client is not mounted on the specified mount directory.
FAULTED	For the local file system, indicates that the block device has unexpectedly unmounted. For the NFS client, indicates that the NFS remote client has unexpectedly unmounted.
UNKNOWN	Indicates that a problem exists either with the configuration or the ability to determine the status of the resource.

Attributes

Table 2-11 Required attributes

Required attribute	Description
BlockDevice	Block device for mount point. When you specify the block device to mount, enclose IPv6 addresses in square brackets. The `mount` command requires square brackets around the IPv6 address to differentiate between the colons in the address and the colon that separates the remote host and remote directory. Type and dimension: string-scalar Examples: ■ "/dev/vx/dsk/myvcs_dg/myvol" ■ IPv4 ■ "10.209.70.90:/dirname/anotherdir" ■ IPv6 "[fe80::1:2:3]:/dirname/anotherdir"
FsckOpt	Mandatory for the following file systems types: ■ jfs ■ jfs2 ■ vxfs Use this attribute to specify options for the `fsck` command. You must correctly set this attribute for local mounts. If the mount process fails, the `fsck` command is executed with the specified options before it attempts to remount the block device. Its value must include either `-y`, `-n`, or `-p`. The -p option is only for jfs or jfs2 file systems on AIX. Refer to the fsck manual page for more information. For NFS mounts, the value of this attribute is not applicable and is ignored. Type and dimension: string-scalar Example: "-n" Example: `"-y"` **Note:** When you use the command line, add the % sign to escape '-'. For example: `hares -modify MntRes FsckOpt %-y`

Table 2-11 Required attributes

Required attribute	Description
FSType	Type of file system. Supports jfs, jfs2, nfs, namefs, or vxfs. Type and dimension: string-scalar Example: "vxfs"
MountPoint	Directory for mount point Type and dimension: string-scalar Example: "/tmp/mnt"

Table 2-11 Required attributes

Required attribute	Description
VxFSMountLock	This attribute is only applicable to Veritas (VxFS) file systems. This attribute controls a file system locking feature to prevent accidental unmounts. This attribute can take three values: 0, 1, or 2. VxFSMountLock=0 The resource does not detect any changes to the lock when VCS reports that it is online after you set the value to zero. ■ If the mount point is initially locked with the mntlock="VCS", the monitor agent function unlocks it. ■ If the mount point is initially locked with a key that is not equal to "VCS", the agent logs a message once. ■ If the mount point is initially not locked, no action is performed. VxFSMountLock=1 The resource does not detect changes to the lock when VCS reports it online after the value was set to one. VCS does not monitor the lock. ■ If the mount point is initially locked with the mntlock="VCS", no action is performed. ■ If the mount point is initially locked with a key that is not equal to "VCS", the agent logs a message once. ■ If the mount point is initially not locked, the monitor agent function locks it with the mntlock="VCS". VxFSMountLock=2 When the value of the VxFSMountLock is 2, the file system is locked and the agent monitors any change to mntlock. ■ If the mount point is locked with the mntlock="VCS", no action is performed. ■ If the mount point is initially locked with a key that is not equal to "VCS", the monitor agent function logs a message whenever a change in mntlock is detected. ■ If the mount point is not locked, the agent locks it with the mntlock="VCS". Type and dimension: integer-scalar Default: 1

Table 2-12 Optional attributes

Optional attribute	Description
MountOpt	Options for the `mount` command. Refer to the `mount` manual page for more information. Do not set the VxFS mount option "mntlock=*key*". The agent uses this option only when bringing a Mount resource online. Type and dimension: string-scalar Example: "rw"
SnapUmount	If the value of this attribute is 1, this attribute automatically unmounts VxFS snapshots when the file system is unmounted. If the value of this attribute is 0, and snapshots are mounted, the resource cannot be brought offline. In this case, failover does not occur. Type and dimension: integer-scalar Default: 0
CkptUmount	If the value of this attribute is 1, this attribute automatically unmounts VxFS Storage Checkpoints when file system is unmounted. If the value of this attribute is 0, and Storage Checkpoints are mounted, then failover does not occur. Type and dimension: integer-scalar Default: 1
SecondLevelMonitor	This attribute has been deprecated. Instead of this attribute, use the LevelTwoMonitorFreq attribute. For more information, see "Enabling second level monitoring for the Mount agent" on page 84.
SecondLevelTimeout	This attribute has been deprecated.

Table 2-12 Optional attributes

Optional attribute	Description
AccessPermissionChk	If the value of this attribute is 1 or 2, the monitor verifies that the values of the MntPtPermission, MntPtOwner, and MntPtGroup attributes are the same as the actual mounted file system values. If any of these do not match the values that you have defined, a message is logged. If the value of this attribute is 2, and if the mounted file system permissions do not match the attribute values, the Monitor function returns the state as OFFLINE. Type and dimension: integer-scalar Default: 0
CreateMntPt	If the value of this attribute is 0, no mount point is created. The mount can fail if the mount point does not exist with suitable permissions. If the value of this attribute is 1 or 2, and a mount point does not exist, the agent creates a mount point with system default permissions when the resource is brought online. If the permissions for the mount point are less than 555, a warning message is logged. If the value of this attribute is 2, and the mount point does not exist, the agent creates a mount point with system default permissions when the resource is brought online. If the permissions for the mount point are less than 555, a warning message is logged. In addition, VCS deletes the mount point and any recursively created directories when the resource is brought offline. The mount point gets deleted only if it is empty, which is also true for recursive mount points. Type and dimension: integer-scalar Default: 0

Table 2-12 Optional attributes

Optional attribute	Description
MntPtGroup	This attribute specifies the group ownership of the mounted file system. The agent verifies the group ownership of the mounted file system every monitor cycle if the value of the AccessPermissionChk attribute is not 0. Type and dimension: string-scalar Example: "grp1"
MntPtOwner	This attribute specifies the user ownership of the mounted file system. The agent verifies the user ownership of the mounted file system every monitor cycle if the value of the AccessPermissionChk attribute is not 0. Type and dimension: string-scalar Example: "usr1"
MntPtPermission	This attribute specifies the permissions of the mounted file system in an absolute format of a four-digit octal. The agent verifies the mode of the mounted file system every monitor cycle if the value of the AccessPermissionChk attribute is not 0. Type and dimension: string-scalar Example: "0755"

Table 2-12 Optional attributes

Optional attribute	Description
OptCheck	The value of this attribute determines if VCS should verify the mount options. The state of the resource is determined based on the result of the verification. If the value of this attribute is 0 (default), the mount options are not checked. If the value of the OptCheck attribute is 1, 2 or 3, a check is performed to see if the mount command options that you have specified for VCS are set in the MountOpt attribute. The MountOpt attributes should be the same as the actual mount command options. If the actual mount options differ from the MountOpt attribute, a message is logged. The state of the resource depends on the value of this attribute. If the value of the attribute is 1, the state of the resource is unaffected. If the value is 2, the state of the resource is set to offline. If the value is 3, state of the resource is set to unknown. Type and dimension: integer-scalar Default: 0
RecursiveMnt	If the value of this attribute is 1, VCS creates all the parent directories of the mount point if necessary. Type and dimension: boolean-scalar Default: 0
ReuseMntPt	If the same mount point needs to be specified in more than one mount resource, set the value of this attribute to 1. Note that this attribute only accepts a value of 1 or 0. To use this attribute, the cluster administrator needs to add this attribute to the ArgList resource type attribute of the agent. Set the appropriate group and resource dependencies such that only one resource can come online on a system at a time. Type and dimension: integer-scalar Default: 0

Resource type definition

```
type Mount (
    static keylist SupportedActions = { "mountpoint.vfd",
    "mounted.vfd", "vxfslic.vfd", "mountentry.vfd", "chgmntlock" }
    static str ArgList[] = { MountPoint, BlockDevice, FSType,
    MountOpt, FsckOpt, SnapUmount, CkptUmount, SecondLevelMonitor,
    SecondLevelTimeout, OptCheck, CreateMntPt, MntPtPermission,
    MntPtOwner, MntPtGroup, AccessPermissionChk, RecursiveMnt,
    VxFSMountLock }
    static int ContainerOpts{} = { RunInContainer=0, PassCInfo=0 }
    static str IMFRegList[] = { MountPoint, BlockDevice, FSType }
    str MountPoint
    str BlockDevice
    str FSType
    str MountOpt
    str FsckOpt
    int SnapUmount = 0
    int CkptUmount = 1
    boolean SecondLevelMonitor = 0
    int SecondLevelTimeout = 30
    int OptCheck = 0
    int CreateMntPt = 0
    int ReuseMntPt = 0
    str MntPtPermission
    str MntPtOwner
    str MntPtGroup
    int AccessPermissionChk = 0
    boolean RecursiveMnt = 0
    int VxFSMountLock = 1
)
```

Mount agent notes

The Mount agent has the following notes:

- “Example 1” on page 83
- “Example 2” on page 83
- “Example 3” on page 83
- “Enabling second level monitoring for the Mount agent” on page 84

High availability fire drill

The high availability fire drill detects discrepancies between the VCS configuration and the underlying infrastructure on a node; discrepancies that might prevent a service group from going online on a specific node. For Mount resources, the high availability drill performs the following, it:

- Checks if the specified mount point directory exists
- Checks if the mount point directory is already used
- Checks for valid Veritas (VxFS) file system licenses
- Checks if the mount point exists in the /etc/filesystems file

For more information about using the high availability fire drill see the *Veritas Cluster Server Administrator’s Guide*.

VxFS file system lock

If the mount option in the mount table output has the option mntlock="*key*", then it is locked with the key "*key*". To verify if mount locking is in use and has the value of "*key*", run the `mount` command and review its output.

```
# mount
```

If the VxFS file system has mntlock="*key*" in its mount options, then unmounting the file system fails.

You can unlock the file system with the `fsadm` command and then unmount it. To unlock a locked mount, run the following command where "*key*" is the lock identifier and *mount_point_name* is the file system mount point.

```
# /opt/VRTS/bin/fsadm -o mntunlock="key" mount_point_name
```

To unmount a file system mounted with locking, run the `umount` command with the option mntunlock=”*key*”, for example:

```
# /opt/VRTS/bin/umount -o mntunlock="key" mount_point_name
```

IMF usage notes

If you use IMF for intelligent resource monitoring, review the following recommendations. Depending on the value of the FSType attribute, you must set the MonitorFreq key value of the IMF attribute as follows:

- FSType attribute value is vxfs:
  - For VxFS version 5.1 SP1:
    You can either set the MonitorFreq to 0 or a high value.
    Setting the value of the MonitorFreq key to a high value will ensure that the agent does not run the monitor function frequently.
    Setting the MonitorFreq key to 0 will disable the traditional monitoring while IMF monitoring is in progress. Traditional monitoring will be done only after receiving the notification for a resource.
    However, if the value of the AccessPermissionChk attribute is set to 1, then set the MonitorFreq key value to the frequency at which you want the agent to run the monitor function.
  - For VxFS versions 5.1 5.0.1 or earlier,
    With VxFS versions prior to 5.1 SP1, VCS IMF only monitors file systems getting mounted and unmounted. To monitor other events, you must enable poll-based monitoring. Set the MonitorFreq key value to the frequency at which you want the agent to run the monitor function.
- FSType attribute value is bindfs:
  IMF registration on Linux for “bind” file system type is not supported.
- In case of SLES11 SP1:
  - IMF should not be enabled for the resources where the BlockDevice can get mounted on multiple MountPoints.
  - If FSType attribute value is nfs, then IMF registration for “nfs” file system type is not supported.

See the *Veritas Cluster Server Administrator’s Guide* for the IMF attribute description.

IPv6 usage notes

Review the following information for IPv6 use:

- For IPv6 functionality for NFS, you must use NFS version 4 in order to make the mount reachable. AIX defaults to NFSv3, which does not work across IPv6. Note that NFSv4 requires several configuration steps in the operating system and NFS-related resources in VCS to enable it on the client and the exporting server.

- Note that AIX's `mount` command refuses to accept IP addresses unless they are resolvable to a hostname.

Bringing a Mount resource online in the WPAR

The Mount resource is brought online in the global environment by default (RunInContainer = 0). If you want to bring a mount resource online inside the WPAR, perform the following:

- Make sure the resource is in a service group that has the ContainerInfo attribute configured.
- Override this attribute at the resource level.
- Set the value of the RunInContainer key to 1.

Selecting the attribute values for a Mount resource for the WPAR's root file system for NFS mounts

For NFS mounts, you can run the SecondLevelMonitor in a container if you configure the following:

- RunInContainer = 0
- PassCInfo = 1
- Use the absolute path for the value of the MountPoint attribute for the Mount resource. The MountPoint attribute should not have the path relative to the WPAR root with this combination.
- Use a value of 1 for the SecondLevelMonitor attribute for the Mount resource.

The following are examples of relative and absolute paths:

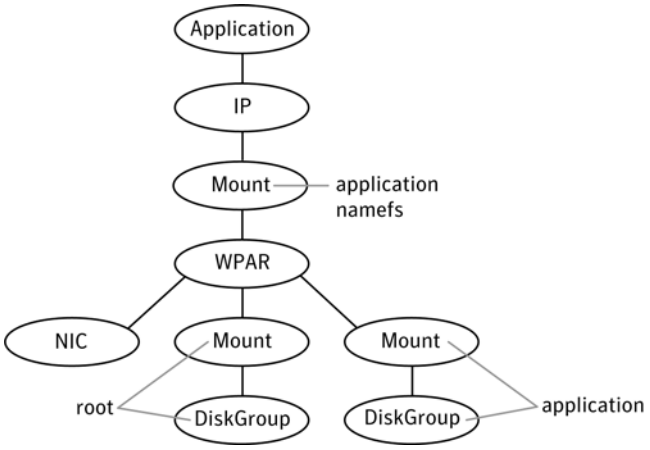
- The file system is mounted on: /wpar/p1/mnt
- The MountPoint attribute's value absolute path: /wpar/p1/mnt
- The MountPoint attribute's value relative path to WPAR root: /mnt

For more information on the ContainerOpts resource attribute, and is RunInContainer and PassCInfo keys, refer to the *Veritas Cluster Server Administrator's Guide*.

Support for namefs file system

The Mount agent provides namefs file system support. You can manage the namefs file system as a Mount resource. Use namefs support to mount a file system in the global environment and share it in the WPAR. For namefs support, configure the FSType attribute to use a value of namefs.

Figure 2-9 Sample service group for the WPAR root on shared storage with a namefs file system when VCS manages the namefs file system as a Mount resource



The following is a sample configuration where you use the Mount resource to manage the namefs file system:

```
group namefssg (
    SystemList = { sysA = 0, sysB = 1 }
    ContainerInfo@sysA = { Name = wpar1, Type = WPAR, Enabled = 1 }
    ContainerInfo@sysB = { Name = wpar1, Type = WPAR, Enabled = 1 }
    )
    Mount namefs_mnt_global_to_local (
                MountPoint = "/wpars/wpar1/namefs_mnt"
                BlockDevice = "/mnt1/m1"
                FSType = namefs
                )
    WPAR w1 (
                )
        Mount base_mnt (
                MountPoint = "/mnt1"
                BlockDevice = "/dev/vx/dsk/tdg/tvol1"
                FSType = vxfs
                FsckOpt = "-y"
                )
    namefs_mnt_global_to_local requires w1
    namefs_mnt_global_to_local requires base_mnt
```

Taking a group with the Mount resource offline can take several minutes if the file system is busy

When a file system has heavy I/O, the `umount` command can take several minutes to respond. However, the `umount` command temporarily deletes the

mount point from mount command output while processing. Per IBM, this is the expected and supported behavior on AIX. The `umount` command's processing later puts the mount point back if the mount point is found busy. Meanwhile, the default OfflineTimeout value of the Mount agent can get exceeded, which in turn invokes the Clean agent function. The Clean function can find the mount point's entry absent from the mount command output and exit with success.

The unmounting, however, may not have happened yet. If unmounting did not occur, offlining resources below the Mount resource (for example the LVMVG or DiskGroup resources) can fail.

The Mount resource's Offline agent function then proceeds to unmount the mount point. After several attempts, the Clean scripts that clean the resources below the Mount resource succeed and the group goes offline.

See the *Veritas Cluster Server Administrator's Guide* for more information about the OfflineTimeout attribute.

Example 1

In this /etc/filesystems entry for a VxFS file system created on a VxVM volume, /mount_point is the mount point for the file system, /dev/vx/dsk/*Diskgroup_name*/*Volume_name* is the block device on which the file system is created, and `vxfs` is the file system type.

```
/etc/filesystems:
/mount_point:
    dev     = /dev/vx/dsk/Diskgroup_name/Volume_name
    vfs     = vxfs    mount    = false
    check   = false
```

Example 2

In this /etc/filesystems entry for a JFS file system created on an LVM logical volume, /mount_point2 is the mount point for the file system, /dev/*LVMlogical_volume* is the block device on which the file system is created, /dev/*LVMlogical_volumelog* is the log device for the file system automatically created by the `crfs` command, and jfs is the file system type.

```
/etc/filesystems:
/mount_point2:
           dev    = /dev/LVMlogical_volume
           vfs    = jfs
           log    = /dev/LVMlogical_volumelog
           mount  = false
           check  = false
```

Example 3

Use the `crfs` and `mkfs` commands to create file systems. VCS supports the following configurations for the Mount agent:

- LVM volume group with a JFS or JFS2 file system.
- VxVM volume with a VxFS file system.

Enabling second level monitoring for the Mount agent

Second level monitoring can be enabled for the Mount agent only if FSType is set to "nfs".

To enable second level monitoring, run the following commands

1. `haconf -makerw`
2. `hares -override` *`resource_name`* `LevelTwoMonitorFreq`
3. `hares -modify` *`resource_name`* `LevelTwoMonitorFreq 1`
4. `haconf -dump -makero`

For more details about the LevelTwoMonitorFreq attribute, refer to the *Veritas Cluster Server Agent Developer's Guide*.

Sample configurations

Configuration 1

In the following configuration, vg00 is a LVM volume group. The mount resource mnt requires the lvmvg_vg00 LVMVG resource.

```
LVMVG lvmvg_vg00 (
    VolumeGroup = vg00
    Disks = { "hdisk3" }
    Options = "u"
)

Mount mnt (
    MountPoint = "/lvm_testmnt"
    BlockDevice = "/dev/lv00"
    FSType = jfs
)
mnt requires lvmvg_vg00
```

Configuration 2

In the following configuration, vol0 is a volume in diskgroup testdg_1 created with VxVM. Mount resource m0 requires the dg1 diskgroup resource.

```
DiskGroup  dg1 (
    DiskGroup = testdg_1
)

Mount m0 (
    MountPoint = "/tmp/m0"
    BlockDevice = "/dev/vx/dsk/testdg_1/vol0"
```

```
    FSType = vxfs
)

m0 requires dg1
```

Configuration 3

In the following configuration, sysA is the remote NFS server and /home/xyz is the remote directory.

```
Mount mnt3 (
    MountPoint = "/tmp/m1"
    BlockDevice = "sysA:/home/xyz"
    FSType = nfs
)
```

Debug log levels

The Mount agent uses the following debug log levels:

DBG_1, DBG_2, DBG_3, DBG_4, DBG_5

Chapter 3

Network agents

This chapter contains the following:

About the network agents

Use network agents to provide high availability for networking resources.

Agent comparisons

IP and NIC agents

The IP and NIC agents:

- Monitor a single NIC
- Support EtherChannel

IPMultiNIC and MultiNICA agents

The IPMultiNIC and MultiNICA agents:

- Monitor single or multiple NICs
- Check the backup NICs at fail over
- Use the original base IP address when failing over
- Provide slower failover compared to MultiNICB but can function with fewer IP addresses
- Have only one active NIC at a time

IPMultiNICB and MultiNICB agents

The IPMultiNICB and MultiNICB agents:

- Monitor single or multiple NICs
- Check the backup NICs as soon as it comes up
- Require a pre-assigned base IP address for each NIC
- Do not fail over the original base IP address
- Provide faster fail over compared to MultiNICA but require more IP addresses
- Have more than one active NIC at a time

802.1Q trunking

The IP/NIC, IPMultiNIC/MultiNICA, and IPMultiNICB/MultiNICB agents support 802.1Q trunking.

To use 802.1Q trunking, create 802.1Q trunked interfaces over a physical interface using SMIT. The physical interface is connected to a 802.1Q trunked port on the switch.

The NIC, MultiNICA, and MultiNICB agents can monitor these trunked interfaces. The IP, IPMultiNIC, and IPMultiNICB agents monitor the virtual IP addresses that are configured on these interfaces.

For example, create a 802.1Q interface called en6 over a physical interface called en0. Do not configure an IP address on en0. You connect en0 to a trunked port on the switch. The NIC and IP agents can then monitor en6 and the virtual IP address configured on en6.

IP agent

The IP agent manages the process of configuring a virtual IP address and its subnet mask on an interface. The virtual IP address must not be in use. You can use this agent when you want to monitor a single IP address on a single adapter.

The interface must be enabled with a physical (or administrative) base IP address before you can assign it a virtual IP address.

For the IP and NIC agents, VCS supports EtherChannel.

This agent is WPAR-aware. The ContainerOpts resource type attribute for this type has a default value of 0 for RunInContainer and a default value of 1 for PassCInfo. Symantec recommends that you do not change these values. Refer to the *Veritas Cluster Server Administrator's Guide* for more information.

High availability fire drill

The high availability fire drill detects discrepancies between the VCS configuration and the underlying infrastructure on a node. These discrepancies might prevent a service group from going online on a specific node. For IP resources, the high availability fire drill:

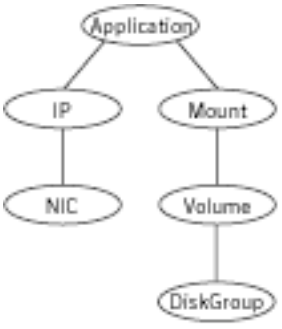
- Checks for the existence of a route to the IP from the specified NIC
- Checks for the existence of the interface configured in the IP and NIC resources

For more information about using the high availability fire drill see the *Veritas Cluster Server Administrator's Guide*.

Dependencies

IP resources depend on NIC resources.

Figure 3-1 Sample service group that includes an IP resource



Agent functions

Online	Uses the `ifconfig` command to set the IP address as an alias on the interface.
Action	The various functions of the action agent are as follows: ■ route.vfd Checks for the existence of a route to the IP from the specified NIC. ■ device.vfd Checks for the existence of the interface configured in the Device attribute.
Offline	Brings down the IP address that is specified in the Address attribute.
Monitor	Monitors the interface to test if the IP address that is associated with the interface is alive.
Clean	Brings down the IP address that is associated with the specified interface.

State definitions

ONLINE	Indicates that the device is up and the specified IP address is assigned to the device.
OFFLINE	Indicates that the device is down or the specified IP address is not assigned to the device.
UNKNOWN	Indicates that the agent could not determine the state of the resource or that the resource attributes are invalid.
FAULTED	Indicates that the IP address could not be brought online, usually because the NIC configured in the IP resource is faulted or the IP address was removed out of VCS control.

Attributes

Table 3-1 Required attributes

Required attribute	Description
Address	A virtual IP address that is different from the base IP address, and that is associated with the interface. Note that the address you specify must not be the same as the configured physical IP address, but should be on the same network. Type and dimension: string-scalar Examples: IPv4: "192.203.47.61" IPv6: "2001::10"
Device	The name of the NIC device that is associated with the IP address. Requires the device name without an alias. Type and dimension: string-scalar Example: "en0"
NetMask	For IPv4 protocol, the subnet mask that is associated with the IP address. Type and dimension: string-scalar Example: "255.255.255.0"
PrefixLen	Required to use the IPv6 protocol. See "PrefixLen" on page 93.

Table 3-2 Optional attributes

Optional attribute	Description
Options	Options for the `ifconfig` command. Type and dimension: string-scalar Example: "metric 4 mtu 1400"

Table 3-2 Optional attributes

Optional attribute	Description
RouteOptions	Specifies the routing options that are passed to the `route add` command when the agent configures an interface. The RouteOptions attribute value is generally formed like this: "*destination gateway metric*". For details about the `route` command, refer to the man page for your operating system. When the value of this string is null, the agent does not add routes. Type and dimension: string-scalar Example: "192.100.201.0 192.100.13.7" In this example, the agent executes the `"route add 192.100.201.0 192.100.13.7"` command when it configures an interface.
PrefixLen	This is the prefix for the IPv6 address represented as the CIDR value. When you use the IPv6 protocol, you must configure values for this attribute and the corresponding NIC agent's Device and Protocol attributes. Type-dimension: integer-scalar Range: 1 - 128 Example: 64

Resource type definition

```
type IP (
    static keylist SupportedActions = { "device.vfd", "route.vfd" }
    static str ArgList[] = { Device, Address, NetMask, Options,
    RouteOptions, PrefixLen }
    static int ContainerOpts{} = { RunInContainer=0, PassCInfo=1 }
    str Device
    str Address
    str NetMask
    str Options
    str RouteOptions
    int PrefixLen
)
```

Sample configurations

NetMask in decimal (base 10)

```
IP IP_192_203_47_61 (
    Device = en0
    Address = "192.203.47.61"
    NetMask = "255.255.248.0"
    )
```

NetMask in hexadecimal (base 16)

```
IP IP_192_203_47_61 (
    Device = en0
    Address = "192.203.47.61"
    NetMask = "0xfffff800"
    )
```

Debug log levels

The IP agent uses the following debug log levels:

DBG_1, DBG_2, DBG_4

NIC agent

The NIC agent monitors the configured NIC. If a network link fails, or if a problem arises with the NIC, the resource is marked FAULTED. You can use the agent to make a single IP address on a single adapter highly available. This resource's Operation value is None.

For the NIC and IP agents, VCS supports EtherChannel.

This agent is compatible with AIX WPARs. The ContainerOpts resource type attribute is not specified for this type. Symantec recommends that you do not change the values for the ContainerOpts keys. Refer to the *Veritas Cluster Server Administrator's Guide* for more information.

High availability fire drill

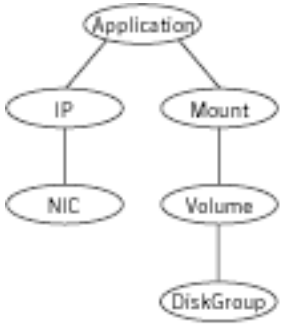
The high availability fire drill detects discrepancies between the VCS configuration and the underlying infrastructure on a node. These discrepancies might prevent a service group from going online on a specific node. For NIC resources, the high availability fire drill checks for the existence of the NIC on the host.

For more information about using the high availability fire drill see the *Veritas Cluster Server Administrator's Guide*.

Dependencies

The NIC resource does not depend on any other resources.

Figure 3-2 Sample service group that includes a NIC resource



The NIC listed in the Device attribute must have an administrative IP address. The administrative IP address is the default IP address that is assigned to the

physical interface of a host on a network. This agent does not configure network routes or administrative IP addresses.

Before you use this agent:

- Verify that the NIC has the correct administrative IP address and subnet mask.
- Verify that the NIC does not have built-in failover support. If it does, disable it.

EtherChannel support

EtherChannel aggregates multiple network interfaces so that they appear as a single interface. For example, you can combine en0 and en1 into an EtherChannel and call the combined interface en2. You then use the NIC agent to monitor this en2 interface. You use the IP agent to configure and monitor an IP address on the en2 interface. Note that you use the en2 interface configured through EtherChannel for the Device attribute.

The IP and NIC agents support EtherChannel use with VCS. EtherChannel is responsible for providing local adapter swapping, which is outside of VCS control. EtherChannel Backup and active-active modes are supported.

Agent functions

Monitor	Tests the network card and network link. Pings the network hosts or broadcast address of the interface to generate traffic on the network. Counts the number of packets passing through the device before and after the address is pinged. If the count decreases or remains the same, the resource is marked FAULTED. If the NetworkHosts list is empty, or the ping test fails, the agent sends a ping to the device's broadcast address to generate network traffic. The agent checks for any response to the broadcast request. If there is no reply to the broadcast ping, the resource faults. Note that for AIX, the systems do not respond to broadcast pings by default. Run the `no -o bcastping=1` command to enable response to broadcast pings.

State definitions

ONLINE	Indicates that the NIC resource is working.
FAULTED	Indicates that the NIC has failed.
UNKNOWN	Indicates the agent cannot determine the interface state. It may be due to an incorrect configuration.

Attributes

Table 3-3 Required attributes

Required attribute	Description
Device	Specifies the name of the NIC that you want to monitor. Use the `lsdev` command to check for all available network adapters. Type and dimension: string-scalar Example: "en0"
NetworkHosts	Required for virtual devices. See "NetworkHosts" on page 98.
Protocol	Required to use the IPv6 protocol. See "Protocol" on page 99.

Table 3-4 Optional attributes

Optional attribute	Description
NetworkHosts	List of hosts on the same network that are pinged to determine if the network connection is alive. Enter the IP address of the host, instead of the host name, to prevent the monitor from timing out. DNS causes the ping to hang. If more than one network host is listed, the monitor returns ONLINE if at least one of the hosts is alive. If you do not specify network hosts, the monitor tests the NIC by sending pings to the broadcast address on the NIC. For a virtual device, you must configure the NetworkHosts attribute. Symantec recommends configuring more than one host to take care of the NetworkHost itself failing. Type and dimension: string-vector Example: { "166.96.15.22", "166.97.1.2" }
NetworkType	Specifies the type of network. Type and Dimension: string-scalar Example: "ether"
PingOptimize	Determines whether to ping every monitor cycle. A value of 0 means that the agent pings either the network host or the broadcast address every monitor cycle. It pings each cycle to determine the state of the network interface. A value of 1 means that the agent uses the device statistics from the netstat output to determine the state of the interface. If no activity exists on the interface, the agent then pings the broadcast address to double-check the state of the network interface. Type and dimension: integer-scalar Default: 1

Table 3-4 Optional attributes

Optional attribute	Description
Protocol	Specifies the type of IP protocol (IPv4 or IPv6) that you want to use with the agent. When you use the IPv6 protocol, you must configure values for this attribute, the Device attribute, and the corresponding IP agent's PrefixLen attribute. Type-dimension: string-scalar Default: IPv4 Example: IPv6

Resource type definition

```
type NIC (
    static keylist SupportedActions = { "device.vfd" }
    static int OfflineMonitorInterval = 60
    static str ArgList[] = { Device, Protocol, PingOptimize,
    NetworkHosts, NetworkType }
    static int ContainerOpts{} = { RunInContainer=0, PassCInfo=0 }
    static str Operations = None
    str Device
    str Protocol = "ipv4"
    int PingOptimize = 1
    str NetworkType
    str NetworkHosts[]
)
```

Sample configurations

Configuration without network hosts (using default ping mechanism)

```
NIC groupx_en0 (
    Device = en0
    PingOptimize = 1
    )
```

Configuration with network hosts

```
NIC groupx_en0 (
    Device = en0
    NetworkHosts = { "10.182.1.1", "10.182.1.2" }
    )
```

IPv6 configuration

The following is a basic configuration for IPv6 with IP and NIC resources. In the following sample, *nic_value* represents the base NIC value for the platform (for example, en0).

```
group nic_group (
    SystemList = { sysA = 0,  sysB  = 1 }
    Parallel = 1
    )

    NIC nic_resource (
        Device@sysA = en0
        Device@sysB = en1
        PingOptimize = 0
        NetworkHosts@sysA = { "2001:db8:c18:2:214:4fff:fe96:11",
        "2001:db8:c18:2:214:4fff:fe96:1" }
        NetworkHosts@sysB = { "2001:db8:c18:2:214:4fff:fe96:1111",
        "2001:db8:c18:2:214:4fff:fe96:111" }
        Protocol = IPv6
        )

    Phantom phantom_resource (
        )

group ip_group (
    SystemList =  { sysA = 0,  sysB  = 1 }
    )

    IP ip_resource (
        Device@sysA = en0
        Device@sysB = en1
        Address = "2001:db8:c18:2:214:4fff:fe96:102"
        PrefixLen = 64
        )

    Proxy proxy_resource (
        TargetResName = nic_resource
        )

    ip_resource requires proxy_resource
```

Debug log levels

The NIC agent uses the following debug log levels:

DBG_1, DBG_2

IPMultiNIC agent

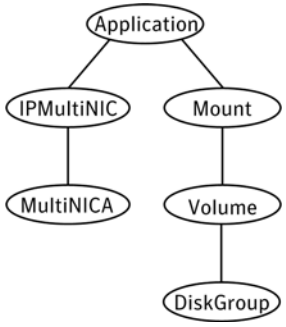
The IPMultiNIC agent manages the virtual IP address that is configured as an alias on one interface of a MultiNICA resource. If the interface faults, the agent works with the MultiNICA resource to fail over to a backup NIC. If multiple service groups have IPMultiNICs associated with the same MultiNICA resource, only one group has the MultiNICA resource. The other groups have Proxy resources pointing to it. You can use this agent for IP addresses on multiple-adapter systems.

The IPMultiNIC and MultiNICA agents supports IPv4 and IPv6.

Dependencies

IPMultiNIC resources depend on MultiNICA resources. They can also depend on WPAR resources.

Figure 3-3 Sample service group that includes an IPMultiNIC resource



Agent functions

Online	Configures a virtual IP address on one interface of the MultiNICA resource.
Offline	Removes the virtual IP address from one interface of the MultiNICA resource.
Monitor	Checks if the virtual IP address is configured on one interface of the MultiNICA resource.

State definitions

ONLINE	Indicates that the specified IP address is assigned to the device.
OFFLINE	Indicates that the specified IP address is not assigned to the device.
UNKNOWN	Indicates that the agent can not determine the state of the resource. This state may be due to an incorrect configuration.
FAULTED	Indicates that the IP address could not be brought online, usually because all the NICs in the MultiNICA resource are faulted or the IP address was removed out of VCS control.

Attributes

Table 3-5 Required attributes

Required attribute	Description
Address	The virtual IP address that is assigned to the active NIC. Type and dimension: string-scalar Examples: IPv4: "10.128.10.14" IPv6: "2001:DB8::"
MultiNICAResName	Name of the associated MultiNICA resource that determines the active NIC. Type and dimension: string-scalar Example: "MultiNICA_res1"
NetMask	The IPv4 protocol netmask for the virtual IP address. Type and dimension: string-scalar Example: "255.255.240.0"
PrefixLen	Required to use the IPv6 protocol. See "PrefixLen" on page 104.

Table 3-6 Optional attributes

Optional attribute	Description
Options	The `ifconfig` command options for the virtual IP address. Type and dimension: string-scalar Example: "mtu 2000"

Table 3-6 Optional attributes

Optional attribute	Description
PrefixLen	Specifies the prefix for the IPv6 address represented as the CIDR value. When you use the IPv6 protocol, you must configure a value for this attribute. Type-dimension: integer-scalar Range: 1 - 128 Example: 64

Note: The default value of the ToleranceLimit static attribute is 3. A value higher than zero helps to prevent the IPMultiNIC agent from performing a failover of the virtual IP address to another system before the MultiNICA agent does a local failover of the virtual IP address.

Resource type definition

```
type IPMultiNIC (
    static str ArgList[] = { "MultiNICAResName:Device", Address,
    NetMask, Options, "MultiNICAResName:Probed",
    "MultiNICAResName:Protocol", MultiNICAResName, PrefixLen }
    static int MonitorTimeout = 120
    static int ToleranceLimit = 3
    str Address
    str NetMask
    str Options
    int PrefixLen
    str MultiNICAResName
)
```

Sample configuration: IPMultiNIC and MultiNICA

Refer to the MultiNICA agent for more information.

```
group grp1 (
    SystemList = { sysa = 0, sysb = 1 }
    AutoStartList = { sysa }
    )
    MultiNICA mnic (
        Device@sysa = { en0 = "10.128.8.42", en1 = "10.128.8.42" }
        Device@sysb = { en0 = "10.128.8.43", en1 = "10.128.8.43" }
```

```
        NetMask = "255.255.255.0"
        Gateway = "10.128.1.1"
        BroadcastAddr = "10.128.8.255"
        )

    IPMultiNIC ip1 (
        Address = "10.128.10.14"
        NetMask = "255.255.255.0"
        MultiNICAResName = mnic
        )

ip1 requires mnic

group grp2 (
    SystemList = { sysa = 0, sysb = 1 }
    AutoStartList = { sysa }
    )

    IPMultiNIC ip2 (
        Address = "10.128.9.4"
        NetMask = "255.255.255.0"
        MultiNICAResName = mnic
        Options = "mtu 1500"
        )

Proxy proxy (
        TargetResName = mnic
        )

ip2 requires proxy
```

MultiNICA agent

The MultiNICA represents a set of network interfaces and provides failover capabilities between them. You can use the agent to make IP addresses on multiple-adapter systems highly available or to monitor them. Each interface in a MultiNICA resource has a base IP address. You can use one base IP address for all NICs, or you can specify a different IP address for use with each NIC. The MultiNICA agent configures one interface at a time. If it does not detect activity on the configured interface, it configures a new interface and migrates IP aliases to it.

If an interface is associated with a MultiNICA resource, do not associate it with any other MultiNICA, MultiNICB, or NIC resource. If the same set of interfaces must be a part of multiple service groups, configure a MultiNICA resource in one of the service groups. Configure the Proxy resources that point to the MultiNICA resource in the other service groups.

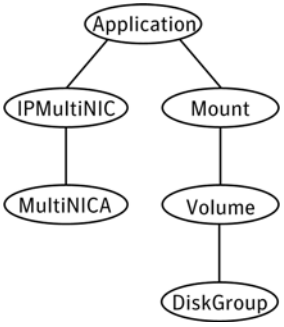
For important information on this agent, refer to:

See “MultiNICA notes” on page 111.

Dependencies

The MultiNICA resource does not depend on any other resources.

Figure 3-4 Sample service group that includes a MultiNICA resource



Agent function

Monitor — Checks the status of the active interface. If the agent detects a failure, it tries to migrate the IP addresses that are configured on that interface. If possible, it tries to migrate the addresses to the next available interface that is configured in the Device attribute.

Note: Systems do not respond to broadcast pings by default. You must run `"no -o bcastping=1"` to enable response to broadcast pings.

State definitions

ONLINE — Indicates that one or more of the network interfaces listed in the Device attribute of the resource is in working condition.

FAULTED — Indicates that all of the network interfaces listed in the Device attribute failed.

UNKNOWN — Indicates that the agent cannot determine the state of the network interfaces that are specified in the Device attribute. This state may be due to incorrect configuration.

Attributes

Table 3-7 Required attributes

Required attribute	Description
BroadcastAddr	Broadcast address Type and dimension: string-scalar Example: "10.192.15.255"
Device	List of interfaces and their base IP addresses. Type and dimension: string-association Example: { en0 = "10.128.8.42", en1 = "10.128.8.42" }

Table 3-7 Required attributes

Required attribute	Description
Gateway	IP address for the default gateway. Type and dimension: string-scalar Example: "10.192.1.7"
NetMask	Netmask for the base IP address. Type and dimension: string-scalar Example: "255.255.255.0"
PrefixLen	Required to use the IPv6 protocol. See "PrefixLen" on page 110.
Protocol	Required to use the IPv6 protocol. See "Protocol" on page 111.

Table 3-8 Optional attributes

Optional attribute	Description
HandshakeInterval	Computes the maximum number of tries the agent makes either to: ■ ping a host (listed in the NetworkHosts attribute) when it fails over to a new NIC, or ■ ping the default broadcast address (depending on the attribute configured) when it fails over to a new NIC. To prevent spurious failovers, the agent must try to contact a host on the network several times before it marks a NIC as FAULTED. Increased values result in longer failover times, whether between the NICs or from system to system in the case of FAULTED NICs. Type and dimension: integer-scalar Default: 1
NetworkHosts	The list of hosts on the network that are pinged to determine if the network connection is alive. Enter the IP address of the host, instead of the host name, to prevent the monitor from timing out. DNS causes the ping to hang. If this attribute is unspecified, the monitor tests the NIC by pinging the broadcast address on the NIC. If more than one network host is listed, the monitor returns online if at least one of the hosts is alive. Type and dimension: string-vector Example: "128.93.2.1", "128.97.1.2"
Options	The `ifconfig` command options for the base IP address. Type and dimension: string-scalar Example: "metric 4 mtu 1400"

Table 3-8 Optional attributes

Optional attribute	Description
PingOptimize	Determines whether to ping every monitor cycle. A value of 0 means that the agent pings either the network host or the broadcast address every monitor cycle. It pings every cycle to determine the state of the network interface. A value of 1 means that the agent uses the device statistics from the netstat output to determine the state of the interface. If no activity exists on the interface, the agent then pings the broadcast address to double-check the state of the network interface. Type and dimension: integer-scalar Default: 1
RouteOptions	String to add a route when configuring an interface. The string contains the destination gateway metric. No routes are added if the value of this string is null. Type and dimension: string-scalar Example: "192.100.201.0 192.100.13.7"
FailoverInProgress	For internal use only.
PrefixLen	Specifies the prefix for the IPv6 address represented as the CIDR value. When you use the IPv6 protocol, you must configure values for this attribute and the MultiNICA agent's Device and Protocol attributes. Type-dimension: integer-scalar Range: 1 - 128 Example: 64

Table 3-8 Optional attributes

Optional attribute	Description
Protocol	Specifies the type of IP protocol (IPv4 or IPv6) that you want to use with the agent. When you use the IPv6 protocol, you must configure values for this attribute, the Device attribute, and the corresponding IPMultiNIC agent's PrefixLen attribute. Type-dimension: string-scalar Default: IPv4 Example: IPv6

Resource type definition

```
type MultiNICA (
    static int OfflineMonitorInterval = 60
    static int MonitorTimeout = 300
    static str ArgList[] = { Device, NetMask, Gateway,
    BroadcastAddr, Options, RouteOptions, PingOptimize,
    MonitorOnly, HandshakeInterval, NetworkHosts, PrefixLen,
    Protocol }
    static str Operations = None
    str Device{}
    str NetMask
    str Gateway
    str BroadcastAddr
    str Options
    str RouteOptions
    int PingOptimize = 1
    int HandshakeInterval = 1
    int PrefixLen
    str Protocol = "ipv4"
    str NetworkHosts[]
    temp boolean FailoverInProgress = 0
)
```

MultiNICA notes

- If all NICs configured in the Device attribute are down, the MultiNICA agent faults the resource after a two-three minute interval. This delay occurs because the MultiNICA agent tests the failed NIC several times before it marks the resource OFFLINE. Failover logs record a detailed description of the events.

- For a single main.cf configuration file, you can only have one MultiNICA resource, which uses either the IPv4 or the IPv6 protocol for a given set of devices. For example, if you have a MultiNICA resource configured as follows:

```
MultiNICA mnic (
    Device@sysa = { en0 = "10.128.8.42", en1 = "10.128.8.42" }
    Device@sysb = { en0 = "10.128.8.43", en1 = "10.128.8.43" }
```

  You cannot have another MultiNICA resource that uses the same device names (en0 and en1) in the main.cf file.

- The MultiNICA agent supports only one active NIC on one IP subnet; the agent does not work with multiple active NICs on the same subnet.
  - On AIX, for example, you have two active NICs, en0 (10.128.2.5) and en1 (10.128.2.8). You configure a third NIC, en2, as the backup NIC to en1. The agent does not fail over from en1 to en2 because some ping tests are redirected through en0 on the same subnet. The redirect makes the MultiNICA monitor return an online status.

EtherChannel support

EtherChannel aggregates multiple network interfaces so that they appear as a single interface. For example you can combine en0 and en1 into an EtherChannel and call the combined interface en2. You then use the MultiNICA agent to monitor this en2 interface. You use the IPMultiNIC agent to configure and monitor an IPMultiNIC address on the en2 interface. Note that you use the en2 interface configured through EtherChannel for the Device attribute.

The IPMultiNIC and MultiNICA bundled agents support EtherChannel use with VCS. EtherChannel is responsible for providing local adapter swapping, which is outside of VCS control. EtherChannel Backup and active-active modes are supported.

Sample configurations

MultiNICA and IPMultiNIC

In the following example, two systems, sysa and sysb, each have a pair of network interfaces, en0 and en1. In this example, the two interfaces, en0 and en1, have the same base, or physical, IP address. Note the lines beginning Device@sysa and Device@sysb; the use of different physical addresses shows how to localize an attribute for a particular host.

The MultiNICA resource fails over the IP addresses to the backup NIC in the event of a failure of the active NIC. The resources ip1 and ip2, shown in the following example, have the Address attribute that contains the logical IP

address. In the event of a NIC failure on sysa, the physical IP address and the two logical IP addresses fails over from en0 to en1.

However, if both the NICs on sysa are disconnected, the MultiNICA and IPMultiNIC resources work in tandem to fault the group on sysa. The entire group now fails over to sysb.

If you have more than one group using the MultiNICA resource, the other groups can use a Proxy resource. The Proxy resource points to the MultiNICA resource in the first group. The Proxy resource prevents redundant monitoring of the NICs on the same system. The IPMultiNIC resource is always made dependent on the MultiNICA resource.

```
group grp1 (
    SystemList = { sysa = 0 , sysb = 1 }
    AutoStartList = { sysa }
    )
    MultiNICA mnic (
        Device@sysa = { en0 = "10.128.8.42", en1 = "10.128.8.42" }
        Device@sysb = { en0 = "10.128.8.43", en1 = "10.128.8.43" }
        NetMask = "255.255.255.0"
        Gateway = "10.128.1.1"
        BroadcastAddr = "10.128.25.255"
        Options = "mtu 1500"
        )

    IPMultiNIC ip1 (
        Address = "10.128.10.14"
        NetMask = "255.255.255.0"
        MultiNICAResName = mnic
        Options = "mtu 1500"
        )

ip1 requires mnic

group grp2 (
        SystemList = { sysa = 0 , sysb = 1 }
        AutoStartList = { sysa }
        )

        IPMultiNIC ip2 (
            Address = "10.128.9.4"
            NetMask = "255.255.255.0"
            MultiNICAResName = mnic
            Options = "mtu 1500"
            )
        Proxy proxy (
            TargetResName = mnic
            )

ip2 requires proxy
```

IPv6 configuration

The following is a basic configuration for IPv6.

```
group mnica_group (

    SystemList = { sysA = 0, sysB = 1 }
    )

    IPMultiNIC ipmnic_res (
        Address = "2007:192::1627:161"
        MultiNICAResName = mnica_res
        PrefixLen = 64
        )

    MultiNICA mnica_res (
        Device@sysA = { en0 = "fe80::214:4fff:fe96:ae0a",
        en1 = "fe80::214:4fff:fe96:ae0a" }
        Device@sysB = { en0 = "fe80::214:4fff:fe98:aeFb",
        en1 = "fe80::214:4fff:fe98:aeFb" }
        PrefixLen = 64
        )

        ipmnic_res requires mnica_res
```

Debug log levels

The MultiNICA agent uses the following debug log levels:

DBG_2, DBG_3, DBG_4, DBG_5

About the IPMultiNICB and MultiNICB agents

The IPMultiNICB and the MultiNICB agents can handle multiple NIC connections. Due to differences in the way that each platform handles its networking connections, these agents vary in design between platforms.

Checklist to ensure the proper operation of MultiNICB

For the MultiNICB agent to function properly, you must satisfy each item in the following list:

- Each interface must have a unique MAC address.
- At boot time, you must configure and connect all the interfaces that are under the MultiNICB resource and give them test IP addresses.
- All test IP addresses for the MultiNICB resource must belong to the same subnet as the virtual IP address.
- If you specify the NetworkHosts attribute, then that host must be on the same subnet as the other IP addresses for the MultiNICB resource.
- If any network host is meant to respond to a broadcast ping, run `no -o bcastping=1` on the network host.
- You must use the AIX SMIT configuration tool to configure the test IP addresses and to make them persistent across reboots. If you do not use SMIT is to configure the IP addresses the agent may failover incorrectly.
- Ensure that media speed settings are the same for both the interface and the corresponding switch port. Symantec recommends setting the media speed to full duplex mode.

IPMultiNICB agent

The IPMultiNICB agent works with the MultiNICB agent. The agent configures and manages virtual IP addresses (IP aliases) on an active network device that the MultiNICB resource specifies. When the MultiNICB agent reports a particular interface as failed, the IPMultiNICB agent moves the IP address to the next active interface. You can use this agent for IP addresses on multiple-adapter systems.

If multiple service groups have IPMultiNICB resources associated with the same MultiNICB resource, only one group should have a MultiNICB resource. The other groups should have a proxy resource pointing to the MultiNICB resource.

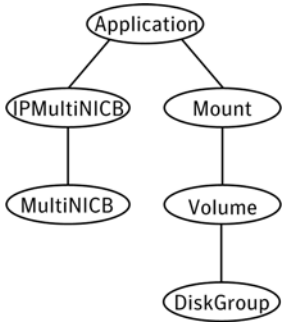
For the MultiNICB and IPMultiNICB agents, VCS supports EtherChannel.

This agent is WPAR-aware. The ContainerOpts resource type attribute for this type has a default value of 0 for RunInContainer and a default value of 1 for PassCInfo. Symantec recommends that you do not change these values. Refer to the *Veritas Cluster Server Administrator's Guide* for more information.

Dependencies

IPMultiNICB resources depend on MultiNICB resources.

Figure 3-5 Sample service group that includes an IPMultiNICB resource



Requirements for IPMultiNICB

The following conditions must exist for the IPMultiNICB agent to function correctly:

- The MultiNICB agent must be running to inform the IPMultiNICB agent of the available interfaces.
- One IPMultiNICB resource can control only one logical IP address.

Minimal configuration

The minimal configuration for this agent consists of:

- The failover IP address.
- The subnet mask.
- The name of the MultiNICB resource that it depends on.

See "Sample configurations" on page 121.

The haipswitch utility

You can use the haipswitch utility to switch IP addresses between MultiNICB interfaces on the same system. Running the utility with the `-h` flag gives an example of usage.

Agent functions

Online	Finds a working interface with the appropriate interface alias or interface name, and configures the logical IP address on it.
Offline	Removes the logical IP address.
Clean	Removes the logical IP address.
Monitor	If the logical IP address is not configured as an alias on one of the working interfaces under a corresponding MultiNICB resource, monitor returns OFFLINE. If the current interface fails, the agent fails over the logical IP address. It fails over the logical IP address to the next available working interface that is within the MultiNICB resource on the same node. If no working interfaces are available then monitor returns OFFLINE.
Open	Data structures necessary for monitoring the network interfaces are created.
Close	Data structures that the monitor agent function uses are freed.
Attr_Changed	Updates the data structures that are used for monitoring the NICs.

State definitions

ONLINE	Indicates that an IP address on one of the working network interfaces of the resource is up. The IP address is specified in the Address attribute. The resource is specified in the MultiNICBResName attribute.
OFFLINE	Indicates that an IP address is not up on any of the working network interfaces of the MultiNICB resource. The IP address is specified in the Address attribute. The resource is specified in the MultiNICBResName attribute.
UNKNOWN	Indicates that the agent cannot determine the status of the virtual IP address that is specified in the Address attribute.
FAULTED	Indicates that the IP address could not be brought online, usually because all the NICs configured in the MultiNICB resource have failed or the IP address was removed out of VCS control.

Attributes

Table 3-9 Required attributes

Required attribute	Description
Address	Defines the dotted decimal failover IP address. This IP address must be different than the base or test IP addresses in the MultiNICB resource. The IPMultiNICB agent automatically assigns the failover IP address. Do not configure this IP address before the IPMultiNICB agent goes online. If the IP address is already configured, the agent returns an error. Type and dimension: string-scalar Example: "10.118.10.15"
MultiNICBResName	Contains the name of the MultiNICB resource that the IPMultiNICB resource depends on. Type and dimension: string-scalar Example: "MultiNICB_res1"
NetMask	The netmask that is associated with the logical IP address. If you do not specify a netmask, the agent uses the operating system's default netmask. Type and dimension: string-scalar Example: "255.255.255.0"
PrefixLen	This is the prefix for the IPv6 address represented as the CIDR value. When you use the IPv6 protocol, you must configure values for this attribute and the corresponding MultiNICB agent's Device and Protocol attributes. Type-dimension: integer-scalar Range: 1 - 128 Example: 64

Table 3-10 Optional attributes

Optional attribute	Description
RouteOptions	Specifies the routing options that are passed to the `route add` command when the agent configures an interface. The RouteOptions attribute value is generally formed like this: `"destination gateway metric"`. For details about the `route` command, refer to the man page for your operating system. When the value of this string is null, the agent does not add routes. Type and dimension: string-scalar Example: "192.100.201.0 192.100.13.7" In this example, the agent executes the `"route add 192.100.201.0 192.100.13.7"` command when it configures an interface.
Options	Options for the `ifconfig` command. Type and dimension: string-scalar Example: "mtu 1500"

Resource type definition

```
type IPMultiNICB (
    static int MonitorTimeout = 120
    static int OfflineMonitorInterval = 60
    static int MonitorInterval = 10
    static str ArgList[] = { Address, NetMask, MultiNICBResName,
    "MultiNICBResName:Probed", RouteOptions, PrefixLen, Options }
    static int ContainerOpts{} = { RunInContainer=0, PassCInfo=1 }
    str Address
    str NetMask
    str MultiNICBResName
    str RouteOptions
    str Options
    int PrefixLen
)
```

Sample configurations

IPMultiNICB and MultiNICB

```
group grp1 (
    SystemList = { sysa = 0 , sysb = 1 }
    AutoStartList = { sysa }
    )

    MultiNICB MNICB_grp1 (
        Device@sysa = { en0 = "10.128.8.42", en1 = "10.128.8.43" }
        Device@sysb = { en0 = "10.128.8.44", en1 = "10.128.8.45" }
        NetworkHosts = "10.128.8.10"
        )

    IPMultiNICB ip1 (
        Address = "10.128.10.14"
        Netmask = "255.255.255.0"
        MultiNICBResName = MNICB_grp1
        )
    ip1 requires MNICB_grp1

group grp2 (
    SystemList = { sysa = 0 , sysb = 1 }
    AutoStartList = { sysa }
    )
    IPMultiNICB ip2 (
        Address = "10.128.10.15"
        Netmask = "255.255.255.0"
        MultiNICBResName = MNICB_grp1
        )
    Proxy MNICB_proxy (
        TargetResName = MNICB_grp1
        )
    ip2 requires MNICB_proxy
```

Other sample configurations for IPMultiNICB and MultiNICB

Refer to the sample configurations in the MultiNICB agent.

Debug log levels

The IPMultiNICB agent uses the following debug log levels:
DBG_1, DBG_4, DBG_5

MultiNICB agent

The MultiNICB agent works with the IPMultiNICB agent. It allows IP addresses to fail over to multiple NICs on the same system before VCS tries to fail over to another system. You can use the agent to make IP addresses on multiple-adapter systems highly available or to monitor them.

When you use the MultiNICB agent, you must configure the NICs before putting them under the agent's control. You must configure all the NICs in a single MultiNICB resource with the IP addresses that are in the same subnet.

You need to set the MONITOR flag for each NIC that the agent controls. Use the `ifconfig` command to set the flag. For example:

```
# ifconfig en0 monitor
```

For the MultiNICB and IPMultiNICB agents, VCS supports EtherChannel.

EtherChannel support

EtherChannel aggregates multiple network interfaces so that they appear as a single interface. For example you can combine en0 and en1 into an EtherChannel and call the combined interface en2. You then use the MultiNICB agent to monitor this en2 interface. You use the IPMultiNICB agent to configure and monitor an IPMultiNICB address on the en2 interface. Note that you use the en2 interface configured through EtherChannel for the Device attribute.

The IPMultiNICB and MultiNICB bundled agents support EtherChannel use with VCS. EtherChannel is responsible for providing local adapter swapping, which is outside of VCS control. EtherChannel Backup and active-active modes are supported.

The haping utility

Use the haping utility (/opt/VRTSvcs/bin/MultiNICB/haping) to test each NIC before you configure the MultiNICB resource. This utility takes the NIC interface as an argument. You can use the utility to perform a link test, a broadcast ping, or to ping a specific remote host. Symantec recommends that the administrator perform a test ping with the remote host before adding it to the NetworkHosts parameter. Note that the remote host should be on the same network as the interface from which you are performing the test ping.

Some examples of the command syntax are as follows:

Link test only on interface en0:

```
haping -l en0
```

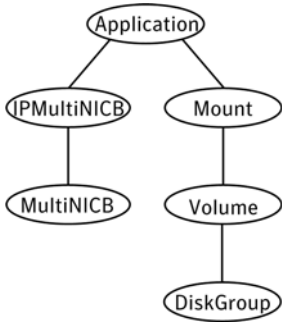
Ping a remote host 10.10.10.10 from interface en0:

```
haping -g 10.10.10.10 en0
```

Dependencies

The MultiNICB resource does not depend on any other resources.

Figure 3-6 Sample service group that includes a MultiNICB resource



Agent functions

Open	Allocates an internal structure to store information about the resource.
Close	Frees the internal structure that is used to store information about the resource.
Monitor	Checks the status of each physical interface. Writes the status information to the export information file for IPMultiNICB resources to read it.

State definitions

ONLINE	Indicates that one or more of the network interfaces listed in the Device attribute of the resource is in working condition.
UNKNOWN	Indicates that the MultiNICB resource is not configured correctly.
FAULTED	Indicates that all of the network interfaces listed in the Device attribute failed.

Attributes

Table 3-11 Required attributes

Required attribute	Description
Device	Lists the interfaces that you want the agent to monitor. You can assign a unique test IP address to each interface. Use the AIX SMIT configuration tool to configure the test IP addresses and to make them persistent across reboots. **Note:** You also must manually configure the default IP route on each NIC in the MultiNICB resource. When you use the IPv6 protocol, you must configure the value for this attribute with base IPv6 addresses. You need to also configure the corresponding IPMultiNICB agent's PrefixLen attribute. Type and dimension: string-association IPv4 example: ■ { en1= "10.182.9.34", en2 = "10.182.10.34" } IPv6 example: ■ { en1 = "2001:db8::1", en2 = "2001:db8::2" }
Gateway	IP address for the default gateway on the local network. Type and dimension: string-scalar Example: "136.22.1.1"

Table 3-12 Optional attributes

Optional attribute	Description
LinkTestRatio	Controls the frequency of the ping test in relation to the link test. The ping test may be run at a lesser frequency to reduce network traffic. If this attribute is set to 1, packets are sent during every monitor cycle. If this attribute is set to 0, packets are never sent during a monitor cycle. Symantec does not recommend setting the value to zero. The agent determines link status without transmitting any ping packets. For other values greater than 1, packets are sent at a lower frequency. For example, if LinkTestRatio=2, then ping packets are sent out during every other monitor cycle. In other words, packets are sent out half as often than if LinkTestRatio were equal to one. Type and dimension: integer-scalar Default: 0
NetworkHosts	The NetworkHosts attribute is a list of hosts on the local network that are pinged to determine if the network connection is available. These must be IP addresses, and not host names. If you do not specify this attribute, the agent monitors the NIC by pinging the broadcast address on the NIC. If you specify one or more network hosts, and at least one host responds to a ping, the agent reports the MultiNICB resource online. The IP addresses for the NetworkHosts attribute must be on the same subnet as the other IP addresses for the MultiNICB resource. If an invalid network host address is specified or if there is mismatch in protocol of the network host and the Protocol attribute of resource, the resource enters an UNKNOWN state. Type and dimension: string-vector Example: "10.128.8.10, 10.128.8.45"

Table 3-12 Optional attributes

Optional attribute	Description
NoBroadcast	If the value of this attribute is 1, NoBroadcast prevents the agent from sending broadcast pings. ARP requests may still be generated. **Note:** If no NetworkHosts are specified and NoBroadcast is set to 1, the agent cannot function properly. Symantec does not recommend setting NoBroadcast to 1. Type and dimension: integer-scalar Default: 0
OfflineTestRepeatCount	Number of times the test is repeated if the interface status changes from up to down. For every repetition of the test, the next NetworkHosts attribute is selected in round-robin manner. At the end of this process, broadcast is performed if NoBroadcast is set to 0. A greater value prevents spurious changes, but increases the response time. Type and dimension: integer-scalar Default: 3
OnlineTestRepeatCount	The number of times that the test is repeated if the interface changes from down to up. This test helps to prevent oscillations in the status of the interface. Type and dimension: integer-scalar Default: 3
NetworkTimeout	Timeout for ARP and ICMP packets in milliseconds. MultiNICB waits for the response to ICMP and ARP packets only during this time period. Assign the NetworkTimeout a value in the order of tens of milliseconds, given that the ICMP and ARP destinations must be on the local network. Increasing this value increases the time for failover. Type and dimension: integer-scalar Default: 100

Resource type definition

```
type MultiNICB (
    static int OfflineMonitorInterval = 60
    static int MonitorInterval = 10
    static str ArgList[] = { Device, NetworkHosts, Gateway,
    LinkTestRatio, NoBroadcast, NetworkTimeout,
    OnlineTestRepeatCount, OfflineTestRepeatCount }
    static str Operations = None
    str Device{}
    str NetworkHosts[]
    str Gateway
    int LinkTestRatio = 0
    int NoBroadcast
    int NetworkTimeout = 100
    int OnlineTestRepeatCount = 3
    int OfflineTestRepeatCount = 3
)
```

Trigger script

MultiNICB monitor agent function calls a VCS trigger in case of an interface going up or down. The agent passes the following arguments to the script:

- MultiNICB resource name
- The device whose status changed, for example:
  - en0
- The device's previous status (0 for down, 1 for up)
- The device's current status and monitor heartbeat

The agent also sends a notification (which may be received via SNMP or SMTP) to indicate that status of an interface changed. The notification is sent using "health of a cluster resource declined" and "health of a cluster resource improved" traps. These traps are mentioned in the *Veritas Cluster Server Administrator's Guide*. A sample mnicb_postchange trigger is provided with the agent. You can customize this sample script as needed or write one from scratch.

The sample script does the following:

- If interface changes status, it prints a message to the console, for example:

  ```
  MultiNICB agent Res. Name: Device en0 status changed from Down
  to Up.
  ```

Sample configurations

IPMultiNICB and MultiNICB configuration

```
group grp1 (
    SystemList = { sysa = 0 , sysb = 1 }
    AutoStartList = { sysa }
    )

    MultiNICB MNICB_grp1 (
        Device@sysa = { en0 = ”10.128.8.42”, en1 = ”10.128.8.43” }
        Device@sysb = { en0 = ”10.128.8.44”, en1 = ”10.128.8.45” }
        NetworkHosts = { “10.128.8.10”, “10.128.8.45” }
        LinkTestRatio = 1
        )

    IPMultiNICB ip1 (
        Address = “10.128.10.14”
        Netmask = “255.255.255.0”
        MultiNICBResName = MNICB_grp1
        )
    ip1 requires MNICB_grp1

group grp2 (
    SystemList = { sysa = 0 , sysb = 1 }
    AutoStartList = { sysa }
    )

    IPMultiNICB ip2 (
        Address = “10.128.10.15”
        Netmask = “255.255.255.0”
        MultiNICBResName = MNICB_grp1
        )
    Proxy MNICB_proxy (
        TargetResName = MNICB_grp1
        )
    ip2 requires MNICB_proxy
```

Debug log levels

The MultiNICB agent uses the following debug log levels:

DBG_1, DBG_2, DBG_3, DBG_4, DBG_5

DNS agent

The DNS agent updates and monitors the mapping for the following:

- The host name to IP address (A, AAAA, or PTR record)
- The canonical name (CNAME)

The agent performs these tasks for a DNS zone when failing over nodes across subnets (a wide-area failover). Resource records (RR) can include different types: A, AAAA, CNAME, and PTR records.

Use the DNS agent when the failover source and target nodes are on different subnets. The agent updates the name server and allows clients to connect to the failed over instance of the application service.

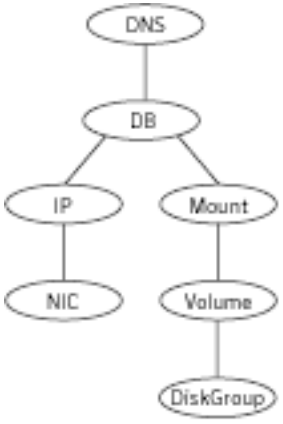
For important information about this agent, refer to:

"DNS agent notes" on page 136

Dependencies

No dependencies exist for the DNS resource.

Figure 3-7 Sample service group that includes a DNS resource



Agent functions

Online

Updates one or more name servers with the resource records.

The agent updates the name servers defined in the StealthMasters attribute. If you have not configured this attribute then the agent obtains the name of the master server by sending an Start of Authority (SOA) query. This query retrieves the SOA record of the zone defined in the agent's Domain attribute. This SOA record contains the name of the master server.

The agent creates PTR records for each RR of type A or AAAA if the value of the CreatePTR attribute is true. A prerequisite for this feature is that the same master or stealth server serves the forward (A or AAAA) and reverse zones.

Finally the agent generates an Online lock file to indicate that the resource is online on the current system.

Offline

Removes the Online lock file.

If attribute OffDelRR is true, offline removes all records that the ResRecord keys define.

Monitor

Returns the ONLINE state if at least one name server reports all mappings that ResRecord defines. The name servers are the master or StealthMaster servers and all the servers for which an NS record for the zone exists.

Clean

Removes the Online lock file, if it exists.

Open

Removes the Online lock file if the resource is reported online on another node inside the cluster to prevent concurrency violation. If the lock file exists, at least one name server has to report all the records that the ResRecord attribute defines. If all the name servers fail to report all the records, the agent function removes the Online lock file.

Action

Different action agent functions follow:

- keyfile.vfd
  This action entry point checks if the key file as specified in the TSIGKeyFile attribute exists either locally or on shared storage.
- dig.vfd
  This action entry point checks if dig and nsupdate binaries exist and are executable.
- master.vfd
  This action entry point checks if stealth masters are able to reply to SOA query for the configured domain.

State definitions

ONLINE	Online lock file exists and at least one name server can return all configured resource records.
OFFLINE	Indicates an offline state when at least one of the following is true: ■ The online lock does not exist. ■ None of the name servers can report all of the RRs' mappings.
UNKNOWN	A problem exists with the configuration. Can indicate that the resource record list contains an invalid value as a part of the record key or a record value of the ResRecord attribute.

Attributes

Table 3-13 Required attributes

Required attribute	Description
Domain	A string representing the DNS zone that the agent administers. The domain name can only contain alphanumeric symbols and the dash. Type and dimension: string-scalar Examples: ■ Forward mapping "demo.example.com" ■ IPv4 reverse mapping "2.168.192.in-addr.arpa"

Table 3-13 Required attributes

Required attribute	Description
ResRecord	ResRecord is an association of DNS resource record values. Each ResRecord attribute consists of two values: *DNS record key* = *DNS record data*. Note that the record key must be a unique value. If the resource record list contains any invalid value as a part of the record key or a record data of the ResRecord attribute, the resource reports an UNKNOWN state. Type and dimension: string-association Examples: ■ For forward mapping, where the zone is demo.example.com: - sles901 = "192.168.2.191" - ww2 = sles901 - sles9ip6 = "2007::1:2:3:abc" ■ For a multi-home DNS record, typically for one host with two network interfaces and different addresses, but the same DNS name. This results in two-A records, or a single A record with continuation lines. sle902 = "192.168.2.102 10.87.13.22" A multi-home AAAA DNS record can be configured as follows: sle902 = "1234::5678 1234::AABB:CCDD" ■ For reverse IPv4 address mapping, where the zone is 2.168.192.in-addr.arpa: 191 = "sles901.demo.example.com." ■ For reverse IPv6 address mapping, where the zone is 3.0.0.0.2.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.7.0.0.2.ip6.arpa: cba = "sles9ip6.demo.example.com." Use only partial host names. If you use a fully qualified domain name, append a period "." at the end of the name. For CNAME records, use: ■ ResRecord = { www = mydesktop } or ■ ResRecord = { www = "mydesktop.marketing.example.com." } Where the Domain attribute is "marketing.example.com"

Table 3-14 Required attributes

Required attribute	Description
ResRecord (cont.)	The agent uses case-insensitive pattern matching—and a combination of the Domain and ResRecord attribute values—to determine the resource record type. The RR type is as follows: ■ PTR: if the Domain attribute ends with .arpa ■ A: if the record data field is four sets of numbers, where a period separates each set. The following details the pattern it tries to match: [*1-223*].[*0-255*].[*0-255*].[*0-255*] Hexadecimal is not supported. ■ AAAA: if the record data fields are in multiple sets of hexadecimal format, then this record is an IPv6 associated type AAAA record. ■ CNAME: for any other valid record data. **Note:** If a name in the ResRecord attribute does not comply with RFC 1035, then a warning is issued to the log file. The ResRecord association is not used. As an exception to this, the DNS agent allows underscore character ("_") in hostnames. Make sure that the DNS server supports the underscore character before you configure any DNS resource records to have the underscore character in their hostnames.

Table 3-15 Optional attributes

Optional attribute	Description
TTL	A non-zero integer represents the “Time To Live” value, in seconds, for the DNS entries in the zone that you want to update. A lower value means more hits on your DNS server, while a higher value means more time for your clients to learn about changes. The time-in-seconds value may take the value 0, which indicates never caching the record, to a maximum of 2,147,483,647, which is over 68 years! The current best practice recommendation (RFC 1912) proposes a value greater than one day, and on RRs that do not change often, consider multi-week values. Type and dimension: integer-scalar Default: 86400 Example: 3600

Table 3-15 Optional attributes

Optional attribute	Description
StealthMasters	The list of primary master name servers in the domain. This attribute is optional since the first name server is retrieved from the zone's SOA (Start of Authority) record. If the primary master name server is a stealth server, define this attribute. A stealth server is a name server that is authoritative for a zone, but does not appear in that zone's SOA record. It is hidden to prevent direct attacks from the Internet. Type and dimension: string-vector Example: { "10.190.112.23" }
TSIGKeyFile	Required when you configure DNS for secure updates. Specifies the absolute path to the file containing the private TSIG (Transaction Signature) key. Type and dimension: string-scalar Example: /var/tsig/example.com.+157+00000.private
CreatePTR	Use the CreatePTR attribute to direct the online agent function to create PTR records for each RR of type A or AAAA. You must set the value of this attribute to true (1) to create the records. Before you can use this attribute, make sure that the same master or stealth servers must serve the forward (A or AAAA) and reverse zones. Type and dimension: boolean-scalar Default: 0 Example: 1
OffDelRR	Use the OffDelRR attribute to direct the offline agent function to remove all records that the ResRecord key defines. You must set the value of this attribute to true (1) to have the agent remove all the records. The online agent function always adds records if they do not exist. Type and dimension: boolean-scalar Default: 0 Example: 1

Resource type definition

```
type DNS (
    static keylist SupportedActions = { "dig.vfd", "master.vfd",
    "keyfile.vfd" }
    static str ArgList[] = { Domain, TTL, TSIGKeyFile,
    StealthMasters, ResRecord, CreatePTR, OffDelRR }
    str Domain
    int TTL = 86400
    str StealthMasters[]
    str TSIGKeyFile
    str ResRecord{}
    boolean CreatePTR = 0
    boolean OffDelRR = 0
)
```

DNS agent notes

The DNS agent has the following notes:

High availability fire drill

The high availability fire drill detects discrepancies between the VCS configuration and the underlying infrastructure on a node; discrepancies that might prevent a service group from going online on a specific node.

For DNS resources, the high availability drill tests the following conditions:

- Checks if the key file as specified by the TSIGKeyFile attribute is available either locally or on shared storage.
- Checks if the dig and nsupdate binaries are available on the cluster node and are executable on that node.
- Checks if the stealth masters can respond to the SOA query made from the cluster node so as to ensure that there is no network issue that would prohibit the DNS update and query requests from reaching the stealth master server.

For more information about using the high availability fire drill see the *Veritas Cluster Server Administrator's Guide*.

Monitor scenarios

Depending on the existence of the Online lock file and the defined Resource Records (RR), you get different status messages from the Monitor function.

Table 3-16 Monitor scenarios for the Online lock file

Online lock file exists	Expected RR mapping	Monitor returns
NO	N/A	OFFLINE
YES	NO	OFFLINE
YES	YES	ONLINE

Sample Web server configuration

Take the former Veritas corporate web server as an example. A browser requests the URL http://www.example.com that maps to the canonical name location1.example.com. The browser retrieves the IP address for the web server by querying a domain name server. If the web server fails over from location one to location two (location2.example.com), the domain name servers need a new canonical name mapping for www.example.com. The www.example.com alias is now updated to point to the canonical name of the standby system in location two.

Secure DNS update for BIND 9

The DNS agent expects that the zone's allow-update field contains the IP address for the hosts that can dynamically update the DNS records. This functionality is default for the DNS agent. Since a competent black hat can, however, spoof IP addresses, consider TSIG as an alternative.

TSIG (Transaction Signature) as specified in RFC 2845 is a shared key message authentication mechanism that is available in DNS. A TSIG key provides the means to authenticate and verify the validity of exchanged DNS data. It uses a shared secret key between a resolver and either one or two servers to provide security.

Setting up secure updates using TSIG keys for BIND 9

In the following example, the domain is example.com.

To use secure updates using TSIG keys

1 Run the `dnssec-keygen` command with the HMAC-MD5 option to generate a pair of files that contain the TSIG key:

```
# dnssec-keygen -a HMAC-MD5 -b 128 -n ZONE veritas.com.
```

2 Open the example.com.+157+00000.key file. After you run the `cat` command, the contents of the file resembles:

```
# cat example.com.+157+00000.key
    example.com. IN KEY 512 3 157 +Cdjlkef9ZTSeixERZ433Q==
```

3 Copy the shared secret (the TSIG key), which looks like:

```
+Cdjlkef9ZTSeixERZ433Q==
```

4 Configure the DNS server to only allow TSIG updates using the generated key. Open the named.conf file and add these lines.

```
key example.com. {
    algorithm hmac-md5;
    secret "+Cdjlkef9ZTSeixERZ433Q==";
};
```

Where **+Cdjlkef9ZTSeixERZ433Q==** is the key.

5 In the named.conf file, edit the appropriate zone section and add the allow-updates sub-statement to reference the key:

```
allow-update { key example.com. ; } ;
```

6 Save and restart the named process.

7 Place the files containing the keys on each of the nodes that is listed in your group's SystemList. The DNS agent uses this key to update the name server. Copy both the private and public key files on to the node. A good location is in the /var/tsig/ directory.

8 Set the TSIGKeyFile attribute for the DNS resource to specify the file containing the private key.

```
DNS www (
Domain = "example.com"
ResRecord = {www = north}
TSIGKeyFile = "/var/tsig/example.com.+157+00000.private"
)
```

Sample configurations

This section contains sample configurations for this agent.

Basic IPv6 configuration

This sample configuration provides basic configuration for IPv6 support. In the following sample, *nic_value* represents the base NIC value for the platform (for example, en0, bge0, eth0, etc.)

```
group ipv6_group_dns (
    SystemList = { sysA = 0, sysB = 1 }
    )

    DNS ipv6group_dns_res (
```

```
        Critical = 0
        Domain = "ipv6.vcs.net"
        TSIGKeyFile =
        "/var/tsig/Kipv6.vcscd.net.+157+18435.private"
        StealthMasters = { "2001:db8:c18:2:69c4:3251:bac1:6cbe" }
        ResRecord = {
            vcssysCv6 = "2001:db8:c18:2:214:4fff:fe96:8833",
            sysC = vcssysCv6 }
        )

    IP ipv6group_ip_res (
        Device @sysA = nic_value
        Device @sysB = nic_value
        Address = "2001:db8:c18:2:214:4fff:fe96:8833"
        PrefixLen = 64
        )

    NIC ipv6group_nic_res (
        Device @sysA = nic_value
        Device @sysB = nic_value
        NetworkHosts = { "2001:db8:c18:2:214:4fff:fea2:fd50" }
        Protocol = IPv6
        )

    ipv6group_dns_res requires ipv6group_ip_res
    ipv6group_ip_res requires ipv6group_nic_res
```

IPv6 CNAME sample configuration

The following sample configuration uses CNAME values.

```
group cname_group (
    SystemList = { sysA = 0, sysB = 1 }
    )

    DNS cname_group_dns_res (
        Domain = "example.com"
        StealthMasters = { "3ffe:556::1000:5761" }
        ResRecord @sysA = { ftp = foo }
        ResRecord @sysB = { ftp = bar }
        OffDelRR = 1
        )
```

IPv4 A sample configuration

The following sample configuration uses A values.

```
group forwardv4_group (
    SystemList = { sysA = 0, sysB = 1 }
    )

    DNS forward_group_v4_resource (
```

```
Domain = "example.com"
StealthMasters = { "3ffe:556::1000:5761" }
ResRecord @sysA = { www = "10.200.56.240" }
ResRecord @sysB = { www = "10.200.56.244" }
OffDelRR = 1
)
```

Debug log levels

The DNS agent uses the following debug log levels:

DBG_1, DBG_2, DBG_3, DBG_4, DBG_5

Chapter 4

File share agents

This chapter contains the following:

About the file service agents

Use the file service agents to provide high availability for file share resources.

NFS agent

Starts and monitors the nfsd and mountd subsystem processes required by all exported NFS file systems.

Note: The attributes NFSv4root and NFSSecurity require AIX 5.3 TL7 SP6 or later and AIX 6.1 TL2 or later.

You should configure only a single NFS resource in a service group on a node. If you have more than one service group that uses the NFS resource, the other service groups must use a Proxy resource. The Proxy resource can point to the NFS resource in the first group. Duplicate NFS resources will cause a problem when the NFS resources are brought online concurrently—only the NFS resource started first will be successfully brought online, while the rest of the NFS resources may report online failure.

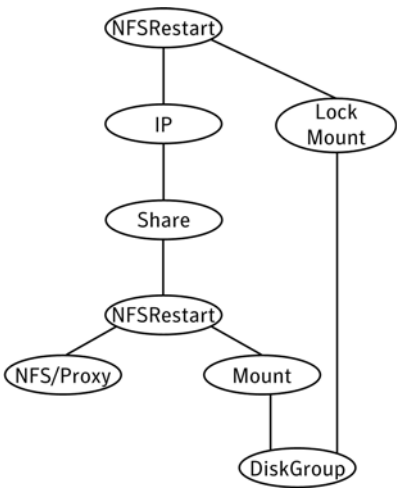
For important information about this agent, refer to:

"NFS agent notes" on page 145

Dependencies

For more information regarding NFS resource dependencies, refer to the *Veritas Cluster Server Administrator's Guide*.

Figure 4-1 Sample service group that includes an NFS resource



Agent functions

Online	■ Checks if nfsd and mountd are running. If they are not running, the agent starts the daemons and exits. ■ The nfsrgyd daemon is started if NFSv4Root is specified. ■ The gssd daemon is started if NFSSecurity is set to 1.
Offline	Not applicable.
Monitor	■ Monitors nfsd and mountd by checking whether or not the daemons are active. ■ The nfsrgyd daemon is monitored if NFSv4Root is specified. ■ The gssd daemon monitored if NFSSecurity is set to 1.
Clean	Terminates the resource and takes it offline—forcibly if necessary.

State definitions

ONLINE	Indicates that the NFS daemons are running in accordance with the supported protocols and versions.
OFFLINE	Indicates that the NFS daemons are not running in accordance with the supported protocols and versions.
FAULTED	Indicates that the NFS daemons are not running in accordance with the supported protocols and versions.
UNKNOWN	Unable to determine the status of the NFS daemons.

Attributes

Table 4-1 Optional attributes

Optional attribute	Description
Nservers	Specifies the number of concurrent NFS requests the server can handle. Type and dimension: integer-scalar Default: 10

Table 4-1 Optional attributes

Optional attribute	Description
NFSv4Root	Root directory of the NFSv4 pseudo file system to be exported. All exports should have a path relative to the path specified by this attribute. You can explicitly create the NFSv4 pseudo file system by specifying the exname option of the `exportfs` command in the Options attribute of the Share resource. If you want to export file systems with NFSv4 protocols and do not want to explicitly create NFSv4 pseudo file system by using the exname option, then set NFSv4Root to "/". Required for filesystems to be exported with v4 protocol. Type and dimension: string-scalar
NFSSecurity	If the value of this attribute is 1,the gssd daemon starts. You must configure the type of security that NFS supports, for example: Kerberos. Type and dimension: boolean-scalar Default: 0
GracePeriod	Specifies the grace period, in seconds, for which the server allows lock recovery. Required for NFS lock recovery. Type and dimension: integer-scalar Default: 90
LockFileTimeout	Specify the amount of time required, in seconds, for the service group to go online. The agent uses this attribute to synchronize the starting and stopping of daemons between multiple service groups. Type and dimension: integer-scalar Default: 180 Example: 240

Resource type definition

```
type NFS (
    static int RestartLimit = 1
    static str ArgList[] = { Nservers, GracePeriod, NFSv4Root,
    NFSSecurity, LockFileTimeout }
    static str Operations = OnOnly
    int Nservers = 10
    int GracePeriod = 90
    str NFSv4Root
    boolean NFSSecurity = 0
    int LockFileTimeout = 180
)
```

NFS agent notes

The NFS agent has the following notes:

Caveat when nodes in the cluster are a mix of AIX operating system versions 5.x and 6.1

Failing over an exported file system between the NFS server nodes with different AIX operating system versions can result in a Stale file handle error at the NFS client. This issue is independent of VCS.

Using NFSv4

For NFS v4 support, you must specify the NFSv4Root attribute. You must include `vers=4` in the Option attribute of the Share resource.

Set up Enterprise Identity Mapping (EIM) in the NFS environment, if:

- Mapping of userids and username is not same on both client and server
- Client and server belong to different domains

If either of the above points are true, and EIM is not set up, the client has minimal rights (user=nobody, group=nobody).

If you want to use the NFSv4 security feature, set the NFSSecurity attribute of the NFS resource to `1`. Manually configure Kerberos or any other security environment that is supported by NFSv4.

Caveats

You export filesystems with NFSv4Root="/exp/exports1", and you forcefully stop the engine so that exports are still valid and existing. If you change

configurations on NFS to set NFSv4Root="/newexport", the NFS Agent is not able to come online with this new root, because the already exported filesystem is using an older NFS pseudo file system root. To avoid this problem bring all Share resources down properly before changing NFSv4Root.

If you create a pseudo file system, a client can access the filesystem. After the NFS server fails over to the other system in the cluster, the client can not see the filesystem. The client needs to remount it.

Sample configurations

On each node in your cluster, you can find sample NFS, NFSRestart, and Share configurations in /etc/VRTSvcs/conf/sample_nfs/.

For more information regarding agent configuration, refer to the *Veritas Cluster Server Administrator's Guide*.

Debug log levels

The NFS agent uses the following debug log levels:

DBG_1, DBG_5

NFSRestart agent

The NFSRestart agent provides the following functionalities:

- Manages essential NFS locking services, network status manager, and lock manager.
- Manages NFS lock recovery service by recovering the NFS record locks after sudden server crash.
- Prevents potential NFS ACK storms by terminating NFS server services before offline of NFS VIP to close all TCP connections with the NFS client.

If you have configured the NFSRestart agent for lock recovery, the NFSRestart agent starts the smsyncd daemon. The daemon copies the NFS locks from the local directory /var/statmon/sm to shared storage. The agent's online function copies the locks from shared storage to local directory /var/statmon/sm.

For important information about this agent, refer to:

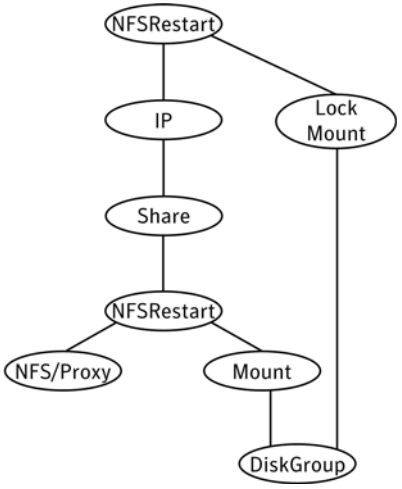
"NFSRestart agent notes" on page 151

Dependencies

For more information regarding NFSRestart resource dependencies, refer to the *Veritas Cluster Server Administrator's Guide*.

You must use two NFSRestart resources in a service group. The lower NFSRestart resource must have its Lower attribute set to 1. The upper NFSRestart resource should be at the top of the resource dependency tree and the lower NFSRestart resource should be below the Share resource in the resource dependency tree. The NFSRestart resources and the Share resources must be inside the same service group.

Figure 4-2 Sample service group that includes an NFSRestart resource



Agent functions

Online

For the lower NFSRestart resource:

- If the value of the NFSLockFailover attribute is 1, the agent terminates statd and lockd.

For the upper NFSRestart resource:

- If the value of the NFSLockFailover attribute is 1, the agent copies the NFS record locks from shared storage to /var/statmon/sm directory.
- Starts the statd and lockd daemons.
- Starts the smsyncd daemon to copy the contents of /var/statmon/sm directory to the shared storage (LocksPathName) at regular two second intervals.

Monitor

For the lower NFSRestart resource:

The monitor agent function does nothing.

For the upper NFSRestart resource:

- If the value of the NFSLockFailover attribute is 1, the agent monitors smsyncd daemon. It restarts the smsyncd daemon if it is not running.
- Monitors the statd and lockd daemons

Offline	For the lower NFSRestart resource: ■ Restarts all the NFS daemons that the upper NFSRestart resource stopped previously. For the upper NFSRestart resource: ■ Terminates the statd and lockd daemons to clear the lock state. ■ Terminates the nfsd and mountd daemons to close the TCP/IP connections. ■ Terminates the smsyncd daemon if the daemon is running.
Clean	For the lower NFSRestart resource: ■ Restarts all the NFS daemons that the upper NFSRestart resource stopped previously. For the upper NFSRestart resource: ■ Terminates the statd and lockd daemons to clear the lock state. ■ Terminates the nfsd and mountd daemons to close the TCP/IP connections. ■ Terminates the smsyncd daemon if the daemon is running.
Action	■ nfsconf.vfd Checks the runlevel information of the system service nfslock to confirm that the lock daemons do not come online automatically after reboot. ■ lockdir.vfd Verifies that the NFS lock directory (which is specified by the LocksPathName attribute of NFSRestart) is on shared storage.

State definitions

ONLINE	Indicates that the daemons are running properly.
OFFLINE	Indicates that one or more daemons are not running.
UNKNOWN	Indicates the inability to determine the agent's status.

Attributes

Table 4-2 Required attributes

Required attribute	Description
NFSRes	Name of the NFS resource on the system. This attribute is required if the value of the NFSLockFailover attribute is 1. Type and dimension: string-scalar

Table 4-3 Optional attributes

Optional attribute	Description
LocksPathName	The path name of the directory to store the NFS lock information. This attribute is required when the value of the NFSLockFailover attribute is 1. The path that you specify for the LocksPathName attribute should be on shared storage. This is to ensure that it is accessible to all the systems where the NFSRestart resource fails over. Type and dimension: string-scalar Example: "/share1x"
NFSLockFailover	A flag that specifies whether the user wants NFS locks to be recovered after a failover. Type and dimension: boolean-scalar Default: 0
Lower	Defines the position of NFSRestart resource in the service group. The NFSRestart resource below the Share resource needs a value of 1. The NFSRestart resource on the top of the resource dependency tree has a Lower attribute value of 0. Type and dimension: integer-scalar Default: 0

Resource type definition

```
type NFSRestart (
    static str ArgList[] = { NFSLockFailover, LocksPathName,
    "NFSRes:GracePeriod", "NFSRes:LockFileTimeout",
    "NFSRes:Nservers", "NFSRes:NFSv4Root", Lower, State }
    static keylist SupportedActions = { "lockdir.vfd", "nfsconf.vfd"
    }
    str LocksPathName
    str NFSRes
    int Lower = 0
    boolean NFSLockFailover = 0
)
```

NFSRestart agent notes

The NFSRestart agent has the following notes:

- "About high availability fire drill" on page 151
- "Providing a fully qualified host name" on page 151

About high availability fire drill

The high availability fire drill detects discrepancies between the VCS configuration and the underlying infrastructure on a node; discrepancies that might prevent a service group from going online on a specific node. For NFSRestart resources, the high availability drill performs the following, it:

- Checks the NFS configuration file to confirm that the NFS server does not come online automatically after reboot.
- Verifies that the NFS lock directory (which is specified by the LocksPathName attribute of NFSRestart) is on shared storage.

For more information about using the high availability fire drill see the *Veritas Cluster Server Administrator's Guide*.

Providing a fully qualified host name

You must provide a fully qualified host name, for example, nfsserver.example.edu, for the NFS server while mounting the file system on the NFS client. If you do not use a fully qualified host name, or if you use a virtual IP address (10.122.12.25) or partial host name (nfsserver), NFS lock recovery may fail.

If you want to use the virtual IP address or a partial host name, make the following changes to the service database (hosts) and the netsvc.conf files:

```
/etc/hosts
```

To use the virtual IP address and partial host name for the NFS server, you need to add an entry to the /etc/hosts file. The virtual IP address and the partial host name should resolve to the fully qualified host name.

```
/etc/netsvc.conf
```

You should also modify the hosts entry in this file so that upon resolving a name locally, the host does not first contact NIS/DNS, but instead immediately returns a successful status. Changing the netsvc.conf file might affect other services running on the system.

For example:

```
hosts = local,bind,nis
```

You have to make sure that the NFS client stores the same information for the NFS server as the client uses while mounting the file system. For example, if the NFS client mounts the file system using fully qualified domain names for the NFS server, then the /var/statmon/sm directory on the NFS client should also contain a fully qualified domain name of the NFS server after the acquisition of locks. Otherwise you need to stop and start the status daemon and lock daemon to clear the lock cache of the NFS client.

A time period exists where the virtual IP address is online but locking services are not registered on the server. Any NFS client trying to acquire a lock in this interval would fail and get ENOLCK error.

Every two seconds, the smsyncd daemon copies the list of clients that hold the locks on the shared filesystem in the service group. If the service group fails before smsyncd has a chance to copy the client list, the clients may not get a notification once the service group is brought up. This causes NFS lock recovery failure.

Sample configurations

On each node in your cluster, you can find sample NFS, NFSRestart, and Share configurations in /etc/VRTSvcs/conf/sample_nfs/.

For more information regarding agent configuration, refer to the *Veritas Cluster Server Administrator's Guide*.

Basic agent configurations

For NFS lock recovery:

```
NFSRestart nfsrestart (
    NFSRes = nfsres
    LocksPathName="/shared_mnt/lockinfo"
    NFSLockFailover = 1
    Lower = 0
    )

NFSRestart nfsrestart_L (
```

```
    NFSRes = nfsres
    LocksPathName="/shared_mnt/lockinfo"
    NFSLockFailover = 1
    Lower = 1
)
```

For no NFS lock recovery:

```
NFSRestart nfsrestart (
    NFSRes = nfsres
    )

NFSRestart nfsrestart_L (
    NFSRes = nfsres
    Lower = 1
    )
```

Debug log levels

The NFSRestart agent uses the following debug log levels:

DBG_1, DBG_3, DBG_4, DBG_5

Share agent

Shares, unshares, and monitors a single local resource for exporting an NFS file system to be mounted by remote systems.

Before you use this agent, verify that the files and directories to be exported are on shared disks.

For important information on this agent, refer to:

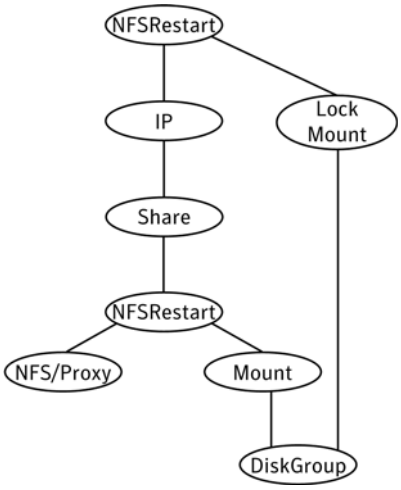
"Share agent notes" on page 157

Dependencies

For more information regarding Share resource dependencies, refer to the *Veritas Cluster Server Administrator's Guide*.

Share resources depend on NFS. In an NFS service group, the IP family of resources depends on Share resources.

Figure 4-3 Sample service group that include a Share resource



Agent functions

Online	Exports (shares) a directory to the specified client.
Offline	Unshares the exported directory from the client.
Monitor	Verifies that the shared directory is exported to the client.
Clean	Terminates all ongoing resource actions and takes the resource offline, forcibly when necessary.
Action	direxists.vfd Checks if the path specified by the PathName attribute exists on the cluster node. If the path name is not specified, it checks if a corresponding mount point is available to ensure that the path is on shared storage.

State definitions

ONLINE	Indicates that specified directory is exported to the client.
OFFLINE	Indicates that the specified directory is not exported to the client.
UNKNOWN	Indicates that the agent could not determine the state of the resource or that the resource attributes are invalid.
FAULTED	Indicates that the share has unexported outside of VCS control.

Attributes

Table 4-4 Required attributes

Required attribute	Description
PathName	Pathname of the file system to be shared. Type and dimension: string-scalar Example: "/share1x"
NFSRes	This attribute has been deprecated.

Table 4-5 Optional attributes

Optional attribute	Description
Options	Options to the `exportfs` command. When specifying multiple options, separate them with commas, for example: "rw,vers=4" For more information about the `exportfs` command and its options, refer to the exportfs manpage. Type and dimension: string-vector

Resource type definition

```
type Share (
    static keylist SupportedActions = { "direxists.vfd" }
    static str ArgList[] = { PathName, Options }
    str PathName
    str Options
)
```

Share agent notes

The following section contains notes on the Share agent.

High availability fire drill

The high availability fire drill detects discrepancies between the VCS configuration and the underlying infrastructure on a node. These discrepancies might prevent a service group from going online on a specific node. For Share resources, the high availability fire drill checks if the path exists.

For more information about using the high availability fire drill see the *Veritas Cluster Server Administrator's Guide*.

Sample configurations

On each node in your cluster, you can find sample NFS, NFSRestart, and Share configurations in /etc/VRTSvcs/conf/sample_nfs/.

For more information regarding agent configuration, refer to the *Veritas Cluster Server Administrator's Guide*.

Debug log levels

The Share agent uses the following debug log levels:

DBG_1

About the Samba agents

Samba is a suite of programs that allows a system running a UNIX or UNIX-like operating system to provide services using the Microsoft network protocol. Samba supports the following services:

- Filespace
- Printer
- WINS
- Domain Master

Configure these services in the Samba configuration file (smb.conf). Samba uses two processes: smbd and nmbd to provide these services.

VCS provides Samba failover using three agents: SambaServer, NetBios, and SambaShare.

The Samba agents

- The NetBios agent
- The SambaServer agent
- The SambaShare agent

Before using the Samba agents

- Verify that smbd and nmbd always run as daemons. Verify that they cannot be started using the meta-daemon inetd.
- Verify that the smbd and nmbd daemons are in the path environment variable.
- The default path of the smbd and nmbd daemons is:
  /usr/local/samba/sbin
  For more information on configuring these paths, refer to the description of the SambaTopDir attribute.
- Verify that Samba is configured properly and that the Samba configuration file is identical on all cluster systems. The user can replicate the file or store it on a shared disk accessible from all cluster systems.
- If configuring Samba as a WINS server or Domain Master, verify that the Samba lock directory is on the shared disk. This ensures that the WINS server database and Domain Master are created on the shared disk.

Supported versions

VCS Samba suite of agents support Samba version 3.0 and above. Please check your samba version using the following command:

```
# smbd -V
```

Note: If you install Samba on AIX 6.1, the AIX 5.3 version is binary compatible.

VCS supports most versions of Samba that are bundled with supported operating systems. For operating systems that do not come bundled with Samba, VCS supports most versions that are compatible with the operating system.

Notes for configuring the Samba agents

The following notes describe configuration considerations for the Samba agents.

Configuring multiple SambaServer resources

For configuring multiple SambaServer resources, configure the SocketAddress attribute with the unique value of the address where the respective samba daemon listens for connections. Configure the SambaServer resource as a parent resource of the IP resource. Configure this IP resource with the SocketAddress attribute value.

Configuring Samba for non-standard configuration files or non-standard lock directories

Configure the PidFile attribute if you use a non-standard configuration file for Samba or if the lock directory (the directory where Samba pid file resides) for Samba is different than the default location. Use the following command to check the standard locations for the Samba configuration file and the lock directory:

To check for the default value of the Samba configuration file

- Enter the following command:

```
# smbd -b | grep CONFIGFILE
```

To check for the default location of the Samba pidfile

- Enter the following command:

```
# smbd -b | grep PIDDIR
```

SambaServer agent

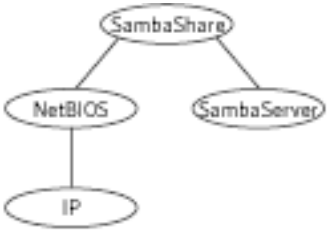
The SambaServer agent starts, stops, and monitors the smbd process as a daemon. You can use the agent to make a smbd daemon highly available.

The smbd daemon provides Samba share services. The agent makes a copy of smbd for each client and verifies that Samba is running by reading the pid of this daemon. The agent can perform in-depth monitoring by establishing a socket connection to Samba at ports where the daemon is listening and sending it a NetBIOS session request.

Dependencies

No dependencies exist for the SambaServer resource.

Figure 4-4 Sample service group that includes a SambaServer resource



Agent functions

Online	Starts the smbd daemon at specified or default ports.
Offline	Stops the smbd daemon.
Monitor	Verifies that the smbd daemon is running by reading its pid file. Does in-depth monitoring periodically, if configured, by establishing a socket connection to Samba and sending it a NetBIOS session request.
Clean	Stops the smbd daemon.

State definitions

ONLINE	Indicates that the smbd daemon is running. If in-depth monitoring is configured, it indicates that a positive session response packet was received through a socket connection to the Samba server.
OFFLINE	Indicates that smbd is not running. If in-depth monitoring is enabled, it indicates that the agent could not establish a socket connection with the server, or that it received an incorrect response packet header, or the session response packet connection timed out.
UNKNOWN	Indicates that the agent could not determine the state of the resource.
FAULTED	Indicates that the smbd daemon has stopped unexpectedly or is not responding (if in-depth monitoring is enabled) outside of VCS control.

Attributes

Table 4-6 Required attributes

Required attribute	Description
ConfFile	Complete path of the configuration file that Samba uses. Type and dimension: string-scalar Example: "/etc/sfw/smb.conf"
LockDir	Lock directory of Samba. Samba stores the files smbd.pid, nmbd.pid, wins.dat (WINS database), and browse.dat (master browser database) in this directory. Type and dimension: string-scalar Example: “ /usr/local/samba/var/locks”
SambaTopDir	Parent path of Samba daemon and binaries. Example: “/usr/local/samba”

Table 4-7 Optional attributes

Optional attribute	Description
IndepthMonitorCyclePeriod	Number of monitor cycles after which the in-depth monitoring is performed. For example, the value 5 indicates that the agent monitors the resource in-depth every five monitor cycles. The value 0 indicates that the agent will not perform in-depth monitoring for the resource. Type and dimension: integer-scalar Default: 5

Table 4-7 Optional attributes

Optional attribute	Description
Ports	Ports where Samba accepts connections. To run Samba over NBT (NetBios over TCP/IP), set this attribute to 139. To run Samba directly over TCP/IP, set this attribute to 445. Type and dimension: integer-vector Default: 139, 445
ResponseTimeout	Number of seconds the agent waits to receive the session response packet after sending the session request packet. For example, the value 5 indicates that the agent waits for five seconds before receiving the session response packet. Configure this attribute if in-depth monitoring is enabled. Type and dimension: integer-scalar Default: 10
PidFile	The absolute path to the Samba daemon pid file. This file contains the process ID of the monitored smbd process. Configure this attribute if you are using a non-standard configuration file name or path. If this agent is not configured for non-standard configuration file names, the agent checks the smbd-*ConfFile*.pid file for monitoring the resource. Type and dimension: string-scalar Example: "/usr/local/samba/var/locks/smbd.pid"
SocketAddress	The IP address where the Samba daemon (smbd) listens for connections. Configure the SocketAddress attribute if you are configuring multiple SambaServer resources on a node. **Note:** Only IPv4 addresses are supported. Type and Dimension: string-scalar Example: “10.128.10.14”

Resource type definitions

```
type SambaServer (
    static str ArgList[] = { ConfFile, SambaTopDir, LockDir, Ports,
    IndepthMonitorCyclePeriod, ResponseTimeout, PidFile,
    SocketAddress }
    str ConfFile
    str LockDir
    int Ports[] = { 139, 445 }
    int IndepthMonitorCyclePeriod = 5
    int ResponseTimeout = 10
    str SambaTopDir
    str PidFile
    str SocketAddress
    )
```

Sample configurations

```
SambaServer samba_server (
    ConfFile = "/etc/smb.conf"
    LockDir = "/usr/local/samba/var/locks"
    SambaTopDir = "/usr/local/samba"
    IndepthMonitorCyclePeriod = 3
    ResponseTimeout = 15
)
```

Debug log levels

The SambaServer agent uses the following debug log levels:

DBG_1, DBG_2, DBG_3, DBG_4, DBG_5

SambaShare agent

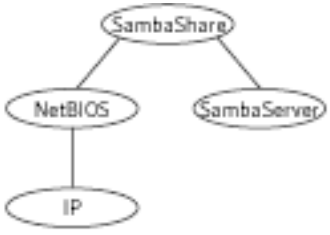
The SambaShare agent adds, removes, and monitors a share by modifying the specified Samba configuration file. You can use the agent to make a Samba Share highly available.

Each filespace or printer service provided by Samba is a shared resource and is defined as a section in the Samba configuration file. The section name is the name of the shared resource and the section parameters define the share attributes.

Dependencies

SambaShare resources depend on the SambaServer, NetBios, and Mount resources.

Figure 4-5 Sample service group for a SambaShare resource



Agent functions

Online	Edits the samba configuration file and adds the shares.
Offline	Removes the shares from the configuration file.
Monitor	Issues the command `smbclient` to check if the specified shares exist.
Clean	Terminates all ongoing resource actions and takes the resource offline, forcibly when necessary.

State definitions

ONLINE	Indicates that the share is available and that the share path exists.

OFFLINE	Indicates that the share is not available, or that the share has a non-existent path.
FAULTED	Indicates that the share has become unavailable outside of VCS control.
UNKNOWN	Indicates that the agent could not determine the state of the resource.

Attributes

Table 4-8 Required attributes

Required attribute	Description
SambaServerRes	Name of the SambaServer resource. Type and dimension: string-scalar Example: "smb_res1"
ShareName	Name of the share resource. Type and dimension: string-scalar Example: "share1"
ShareOptions	List of parameters for the share attributes. These parameters are specified as name=value pairs, with each pair separated by a semicolon (;). Type and dimension: string-scalar Example: "path=/shared; public=yes; writable=yes"

Resource type definition

```
type SambaShare (
    static str ArgList[] = { "SambaServerRes:ConfFile",
    "SambaServerRes:SambaTopDir", "SambaServerRes:LockDir",
    ShareName, ShareOptions, "SambaServerRes:Ports",
    SambaServerRes, "SambaServerRes:PidFile",
    "SambaServerRes:SocketAddress" }
    str SambaServerRes
    str ShareName
    str ShareOptions
    )
```

Sample configuration

```
SambaShare Samba_SambaShare3 (
        SambaServerRes = Samba_SambaServer
        ShareName = smbshare3
        ShareOptions = "path=/smbshare3; public=yes; writable=yes"
        )
```

Debug log levels

The SambaShare agent uses the following debug log levels:

DBG_1, DBG_5

NetBios agent

The NetBios agent starts, stops, and monitors the nmbd daemon. You can use the agent to make the nmbd daemon highly available.

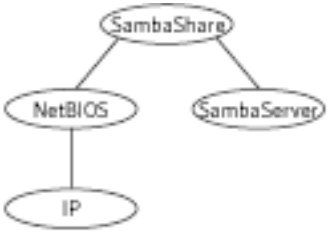
The agent sets, monitors, and resets the names and network interfaces by which the Samba server is known. The agent also sets, monitors and resets Samba to act as a WINS server or domain master or both.

Note that nmbd broadcasts the NetBIOS name, or the name by which the Samba server is known in the network.

Dependencies

The NetBios resource depends on the IP, IPMultiNIC, or IPMultiNICB resource if the virtual IP address configured in the IP/IPMultiNIC resource is being used in the Interfaces attribute of the NetBios resource.

Figure 4-6 Sample service group that includes a NetBIOS resource



Agent functions

Online	Updates the Samba configuration with the NetBIOS name, all NetBIOS aliases and network interfaces, WINS support, and domain master options specified in the NetBIOS resource. Starts the nmbd daemon.
Offline	Removes the NetBIOS name, all NetBIOS aliases and network interfaces, WINS support, and domain master options specified in the NetBIOS resource from the Samba configuration file. Stops the nmbd daemon.
Monitor	Verifies that the Samba configuration contains the NetBIOS name, all NetBIOS aliases and network interfaces, WINS support, and domain master options specified in the NetBIOS resource.
Clean	Terminates all ongoing resource actions and takes the resource offline, forcibly when necessary.

State definitions

ONLINE — Indicates that the specified NetBIOS aliases are advertised and that Samba is handling requests for all specified network interfaces. Indicates that WINS and Domain support services are running, if configured.

OFFLINE — Indicates one or more of the following:

- NetBIOS name is not advertised.
- A NetBIOS alias is not advertised.
- Samba is not handling requests on one of the specified interfaces.
- If WINS support is configured, Samba is not providing WINS service.
- If domain support is set, Samba is not providing Domain Master service.

UNKNOWN — Indicates that the agent could not determine the state of the resource.

FAULTED — Indicates that the nmbd daemon has stopped unexpectedly outside of VCS control.

Attributes

Table 4-9 Required attributes

Required attribute	Description
NetBiosName	Name by which the Samba server is known in the network. Type and dimension: string-scalar
SambaServerRes	Name of the SambaServer resource. Type and dimension: string-scalar Example: "smb_res1"

Table 4-10 Optional attributes

Optional attribute	Description
Interfaces	List of network interfaces on which Samba handles browsing. Type and dimension: string-vector Example: "172.29.9.24/16" **Note:** Note that if you have configured the SocketAddress attribute value for the corresponding SambaServer resource, then you must also configure the same value paired with the appropriate netmask in the list of interfaces.
NetBiosAliases	List of additional names by which the Samba server is known in the network. Type and dimension: string-vector Example: "host1_samba, myname"
WinsSupport	If set to 1, this flag causes the agent to configure Samba as a WINS server. Type and dimension: integer-scalar Default: 0
DomainMaster	If the value of this attribute is 1, the agent sets Samba as Domain Master. Type and dimension: integer-scalar Default: 0

Table 4-10 Optional attributes

Optional attribute	Description
PidFile	The absolute path to the NetBIOS daemon pid file. This file contains the process ID of the monitored nmbd process. Configure this attribute if you are using a non-standard configuration file name or path. If this agent is not configured for non-standard configuration file names, the agent checks for the nmbd-*ConfFile*.pid file for resource monitoring. Type and dimension: string-scalar Example: "/usr/local/samba/var/locks/nmbd.pid"

Resource type definition

```
type NetBios (
    static str ArgList[] = { "SambaServerRes:ConfFile",
    "SambaServerRes:SambaTopDir", "SambaServerRes:LockDir",
    NetBiosName, NetBiosAliases, Interfaces, WinsSupport,
    DomainMaster, "SambaServerRes:PidFile", SambaServerRes,
    PidFile }
    str SambaServerRes
    str NetBiosName
    str NetBiosAliases[]
    str Interfaces[]
    int WinsSupport
    int DomainMaster
    str PidFile
    )
```

Sample configuration

```
NetBios Samba_NetBios (
    SambaServerRes = Samba_SambaServer
    NetBiosName = samba_demon
    NetBiosAliases = { asamba_demon, samba127 }
    WinsSupport = 1
    DomainMaster = 1
    )
```

Debug log levels

The NetBios agent uses the following debug log levels:

DBG_1, DBG_5

Chapter 5

Service and application agents

This chapter contains the following agents:

About the service and application agents

Use service and application agents to provide high availability for application and process-related resources.

Apache Web server agent

The Apache Web server agent brings an Apache Server online, takes it offline, and monitors its processes. The Apache Web server agent consists of resource type declarations and agent scripts. You use the Apache Web server agent, in conjunction with other agents, to make an Apache Web server highly available.

This agent supports the Apache HTTP server 1.3, 2.0, and 2.2. It also supports the IBM HTTP Server 1.3 and 2.0.

This agent can detect when an Apache Web server is brought down gracefully by an administrator. When Apache is brought down gracefully, the agent does not trigger a resource fault even though Apache is down.

Note: The Apache agent requires an IP resource for operation.

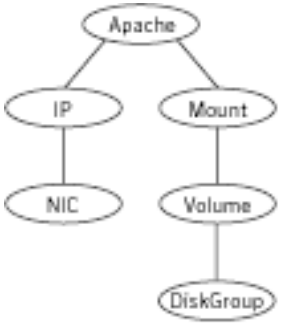
For more information regarding this agent:

See "Apache Web server notes" on page 181.

Dependencies

This type of resource depends on IP and Mount resources.

Figure 5-1 Sample service group for the Apache Web server agent



Agent functions

Online	Starts an Apache server by executing the httpdDir/httpd program with the appropriate arguments. When you specify a file with the EnvFile attribute, the file is sourced before the agent executes the httpd command.
Offline	To stop the Apache HTTP server, the agent: ■ Executes the httpdDir/httpd program with the appropriate arguments (Apache v2.0), or ■ Sends a TERM signal to the HTTP Server parent process (Apache v1.3). When you specify a file with the EnvFile attribute, the file is sourced before the agent executes the httpd command.
Monitor	Monitors the state of the Apache server. First it checks for the processes, next it can perform an optional state check.
Clean	Removes the Apache HTTP server system resources that might remain after a server fault or after an unsuccessful attempt to online or offline. These resources include the parent httpd daemon and its child daemons.
Action	checkconffile.vfd Checks for the existence of the Apache configuration file and the existence of the directory that contains the httpd binary that is used during start up. For a local installation, if the config file or HttpdDir is not found, make sure that it exists on the failover node.

State definitions

ONLINE	Indicates that the Apache server is running.
OFFLINE	Indicates that the Apache server is not running. Can also indicate that the administrator has stopped the Web server gracefully. Note that the agent uses the PidFile attribute for intentional offline detection.
UNKNOWN	Indicates that a problem exists with the configuration.

Attributes

Table 5-1 Required attributes

Required attribute	Description
ConfigFile	Full path and file name of the main configuration file for the Apache server. Type and dimension: string-scalar Example: "/apache/server1/conf/httpd.conf"
httpdDir	Full path of the directory to the httpd binary file Type and dimension: string-scalar Example: "/apache/server1/bin"
ResLogLevel	Controls the agent's logging detail for a specific instance of a resource. Values are: ■ ERROR: Logs error messages. ■ WARN: Logs error and warning messages. ■ INFO: Logs error, warning, and informational messages. ■ TRACE: Logs error, warning, informational, and trace messages. Trace logging is verbose. Use for initial configuration or troubleshooting. Type and dimension: string-scalar Default: INFO Example: "TRACE"
PidFile	This attribute is required when you want to enable the detection of a graceful shutdown outside of VCS control. See "PidFile" on page 179.
EnvFile	This attribute may be required when you use IBM HTTP Server. See "EnvFile" on page 178.

Table 5-2 Optional attributes

Optional attribute	Description
DirectiveAfter	A list of directives that httpd processes after reading the configuration file. Type and dimension: string-association Example: DirectiveAfter{} = { KeepAlive=On }
DirectiveBefore	A list of directives that httpd processes before it reads the configuration file. Type and dimension: string-association Example: DirectiveBefore{} = { User=nobody, Group=nobody }
User	Account name the agent uses to execute the httpd program. If you do not specify this value, the agent executes httpd as the root user. Type and dimension: string-scalar Example: "apache1"
EnableSSL	Set to 1 (true) to have the online agent function add support for SSL by including the option `-DSSL` in the start command. For example: `/usr/sbin/httpd -f path_to_httpd.conf -k start -DSSL` Where path_to_httpd.conf file is the path to the httpd.conf file. Set to 0 (false) it excludes the `-DSSL` option from the command. Type and dimension: boolean-scalar Default: 0 Example: "1"

Table 5-2 Optional attributes

Optional attribute	Description
HostName	The virtual host name that is assigned to the Apache server instance. The host name is used in second-level monitoring for benchmarking the Apache HTTP server. You can use IPv4 or IPv6 addresses for the HostName attribute. **Note:** The HostName attribute is only required when the value of SecondLevelMonitor is 1 (true). Type and dimension: string-scalar Example: "web1.example.com"
Port	Port number where the Apache HTTP server instance listens. The port number is used in second-level monitoring for benchmarking the Apache HTTP server. Specify this attribute only if SecondLevelMonitor is set to 1 (true). Type and dimension: integer-scalar Default: 80 Example: "80"
EnvFile	Full path and file name of the file that is sourced before executing httpdDir/httpd. With Apache 2.0, the file *ServerRoot*/bin/envvars, which is supplied in most Apache 2.0 distributions, is commonly used to set the environment before executing httpd. Specifying this attribute is optional. If EnvFile is specified, the shell for user root must be Bourne, Korn, or C shell. This attribute may be required when you use the IBM HTTP Server if the online action fails. For example: set the EnvFile to /usr/IBM/HTTPServer/bin/envvars. Type and dimension: string-scalar Example: "/apache/server1/bin/envvars"

Table 5-2 Optional attributes

Optional attribute	Description
PidFile	The PidFile attribute sets the file to which the server records the process ID of the daemon. The value of PidFile attribute must be the absolute path where the Apache instance records the pid. This attribute is required when you want the agent to detect the graceful shutdown of the Web server. For the agent to detect the graceful shutdown of the Web server, the value of the IntentionalOffline resource type attribute must be 1 (true). Type and dimension: string-scalar Example: /var/run/httpd.pid
SharedObjDir	Full path of the directory in which the Apache HTTP shared object files are located. Specifying this attribute is optional. It is used when the HTTP Server is compiled using the SHARED_CORE rule. If you specify this attribute, the directory is passed to the `-R` option when executing the httpd program. Refer to the httpd man pages for more information about the `-R` option. Type and dimension: boolean-scalar Example: "/apache/server1/libexec"
SecondLevelMonitor	Enables second-level monitoring for the resource. Second-level monitoring is a deeper, more thorough state check of the Apache HTTP server. Valid attribute values are 1 (true) and 0 (false). Specifying this attribute is required. Type and dimension: boolean-scalar Default: 0 Example: "1"

Table 5-2 Optional attributes

Optional attribute	Description
SecondLevelTimeout	The number of seconds that the monitor agent function waits on the execution of second-level monitor. If the second-level monitor program does not return to calling the monitor agent function before the SecondLevelTimeout window expires, the monitor agent function no longer blocks on the program sub-process. It does, however, report that the resource is offline. The value should be high enough to allow the second level monitor enough time to complete. The value should be less than the value of the agent's MonitorTimeout. Type and dimension: integer-scalar Default: 30

Table 5-3 Resource type attribute

Required attribute	Description
IntentionalOffline	For information on how to use the IntentionalOffline resource type attribute, refer to the *Veritas Cluster Server Administrator's Guide*.

Resource type definition

```
type Apache (
    static keylist SupportedActions = { "checkconffile.vfd" }
    static str ArgList[] = { ResLogLevel, State, IState, httpdDir,
    SharedObjDir, EnvFile, PidFile, HostName, Port, User,
    SecondLevelMonitor, SecondLevelTimeout, ConfigFile, EnableSSL,
    DirectiveAfter, DirectiveBefore }
    str ResLogLevel = INFO
    str httpdDir
    str SharedObjDir
    str EnvFile
    str PidFile
    str HostName
    int Port = 80
    str User
    boolean SecondLevelMonitor
    int SecondLevelTimeout = 30
    str ConfigFile
    boolean EnableSSL
    str DirectiveAfter{}
```

```
    str DirectiveBefore{}
    static boolean IntentionalOffline = 0
)
```

Apache Web server notes

The Apache Web server has the following notes:

- "Tasks to perform before you use the Apache Web server agent" on page 181
- "About detecting application failure" on page 182
- "About bringing an Apache Web server online outside of VCS control" on page 182
- "About high Availability fire drill" on page 182

Tasks to perform before you use the Apache Web server agent

Before you use this agent, perform the following tasks:

- Install the Apache server on shared or local disks.
- Ensure that you are able to start the Apache Web server outside of VCS control, with the specified parameters in the Apache configuration file (for example: /etc/apache/httpd.conf). For more information on how to start the server:
  See "About bringing an Apache Web server online outside of VCS control" on page 182.
- Specify the location of the error log file in the Apache configuration file for your convenience (for example: ErrorLog /var/apache/logs/error_log).
- Verify that the floating IP has the same subnet as the cluster systems.
- If you use a port other than the default 80, assign an exclusive port for the Apache server.
- Verify that the Apache server configuration files are identical on all cluster systems.
- Verify that the Apache server does not autostart on system startup.
- Verify that `Inetd` does not invoke the Apache server.
- Remove previous versions of this agent.
- The service group has disk and network resources to support the Apache server resource.
- Assign virtual host name and port to Apache Server.

About detecting application failure

The agent provides two methods to evaluate the state of an Apache HTTP server instance. The first state check is mandatory and the second is optional.

The first check determines the state of the Apache HTTP server. The check determines the state by searching for the existence of the parent httpd daemon. It also searches for at least one child httpd daemon. If the parent process and at least one child do not exist, VCS reports the resource as offline. If they do exist, and if the agent attribute SecondLevelMonitor is set to true, then the Apache agent uses the Apache Benchmarking utility "ab" to perform detail monitoring. If the exit code of the "ab" utility is 0 and if the command output contains "Benchmarking *HostName*", the agent considers the server online, else the agent considers the server offline.

About bringing an Apache Web server online outside of VCS control

When you bring an Apache Web server online outside of VCS control, first source its environment file. Start the server with the `-f` option so the server knows which instance to start. You can then specify additional options (such as EnableSSL or SharedObjDir) that you want the server to use at start.

To start an Apache Web server outside of VCS control

1. Source the environment file if required.
2. Start the Apache Web server. You must use the `-f` option so that the agent can distinguish different instances of the server.
   httpdDir`/httpd -f` *ConfigFile* `-k start`
   Where *httpdDir* is /apache/v2.2/bin *ConfigFile* is /apache/v2.2/conf/httpd.conf. When fully formed, the start example looks like:
   `/apache/v2.2/bin/httpd -f /apache/v2.2/conf/httpd.conf -k start`
3. Specify additional options such as EnableSSL or SharedObjDir that you want to use when you start server. When you add EnableSSL to the command, it resembles:
   httpdDir`/httpd -f` *ConfigFile* `-k start -DSSL`

About high Availability fire drill

The high availability fire drill detects discrepancies between the VCS configuration and the underlying infrastructure on a node. These discrepancies might prevent a service group from going online on a specific node. For Apache resources, when the Apache Web server is installed locally, the high availability fire drill checks for the validity of these attributes:

- ConfigFile

- httpdDir

For more information about using the high availability fire drill see the *Veritas Cluster Server Administrator's Guide*.

Sample configurations

The following is a basic configuration for the resource.

```
group ApacheG1(
        SystemList = { host1 = 0, host2 = 1 }
        )

        Apache httpd_server (
                Critical = 0
                httpdDir = "/apache/bin"
                HostName = vcsaix1
                Port = 8888
                User = root
                SecondLevelMonitor = 1
                ConfigFile = "/apache/conf/httpd.conf"
                )

        DiskGroup Apache_dg (
                Critical = 0
                DiskGroup = apc1
                )

        IP Apache_ip (
                Critical = 0
                Device = en0
                Address = "11.123.99.168"
                NetMask = "255.255.254.0"
                )

        Mount Apache_mnt (
                Critical = 0
                MountPoint = "/apache"
                BlockDevice = "/dev/vx/dsk/apc1/apcvol1"
                FSType = vxfs
                FsckOpt = "-y"
                )

        Apache_mnt requires Apache_dg
        httpd_server requires Apache_mnt
        httpd_server requires Apache_ip
```

Basic IPv6 configuration

The following is a basic IPv6 configuration for the resource.

```
group ipv6group (
    SystemList = { sysA = 0, sysB = 1 }
    )

    Apache ipv6group_apache_res (
        HostName = "fd4b:454e:205a:110:211:25ff:fe7e:118"
        PidFile = "/myapache/apache/logs/httpd.pid"
        httpdDir = "/myapache/apache/bin"
        ConfigFile = "/myapache/apache/conf/httpd.conf"
        ResLogLevel = TRACE
        SecondLevelTimeout = 20
        IntentionalOffline = 1
        )

    DiskGroup ipv6group_dg_res (
        DiskGroup = dg01
        )

    IP ipv6group_ip_res (
        Device = en0
        Address = "fd4b:454e:205a:110:211:25ff:fe7e:118"
        PrefixLen = 64
        )

    Mount ipv6group_mnt_res (
        MountOpt = rw
        FsckOpt = "-n"
        BlockDevice = "/dev/vx/dsk/dg01/vol01"
        MountPoint = "/myapache/apache"
        FSType = vxfs
        )

    NIC ipv6group_nic_res (
        Device = en0
        )

    Volume ipv6group_vol_res (
        Volume = vol01
        DiskGroup = dg01
        )

    ipv6group_apache_res requires ipv6group_mnt_res
    ipv6group_apache_res requires ipv6group_ip_res
    ipv6group_mnt_res requires ipv6group_vol_res
    ipv6group_vol_res requires ipv6group_dg_res
    ipv6group_ip_res requires ipv6group_nic_res
```

Application agent

The Application agent brings applications online, takes them offline, and monitors their status. Use it to specify different executables for the online, offline, and monitor routines for different programs. The executables must exist locally on each node. You can use this agent to provide high availability for applications that do not have bundled, enterprise, or custom agents.

An application runs in the default context of root. Specify the user name to run an application in a user context.

You can monitor the application in the following ways:

- Use the monitor program
- Specify a list of processes
- Specify a list of process ID files
- Any combination of the above

This agent is WPAR-aware. The ContainerOpts resource type attribute for this type has a default value of 1 for RunInContainer and a default value of 0 for PassCInfo. Symantec recommends that you do not change these values. Refer to the *Veritas Cluster Server Administrator's Guide* for more information.

This agent is IMF-aware and uses asynchronous monitoring framework (AMF) kernel driver for IMF notification. For more information about the Intelligent Monitoring Framework (IMF) and intelligent resource monitoring, refer to the *Veritas Cluster Server Administrator's Guide*.

High availability fire drill

The high availability fire drill detects discrepancies between the VCS configuration and the underlying infrastructure on a node. These discrepancies might prevent a service group from going online on a specific node. For Application resources, the high availability fire drill checks for:

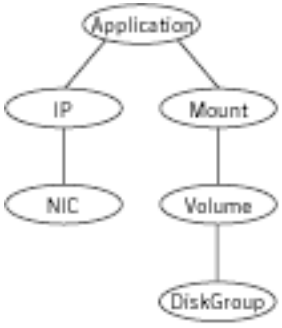
- The availability of the specified program (program.vfd)
- Execution permissions for the specified program (program.vfd)
- The existence of the specified user on the host (user.vfd)
- The existence of the same binary on all nodes (cksum.vfd)

For more information about using the high availability fire drill see the *Veritas Cluster Server Administrator's Guide*.

Dependencies

Depending on how you plan to use it, an Application type of resource can depend on IP and Mount resources. Alternatively, instead of the IP resource you can also use the IPMultiNIC or IPMultiNICB resource.

Figure 5-2 Sample service group that includes an Application resource



Agent functions

Online Runs the command or script that you specify in the value of the StartProgram attribute. Runs the command with the specified parameters in the context of the specified user.

To bring the resource online, the agent function performs the command`su [-] user -c command_to_online_resource`

Offline Runs the command or script that you specify in the value of the StopProgram attribute. Runs the command with the specified parameters in the context of the specified user.

To take the resource offline, the agent function performs the command:

```
su [-] user -c command_to_offline_resource
```

Monitor If you specify the MonitorProgram attribute, the agent executes the user-defined MonitorProgram in the user-specified context. If you specify the PidFiles attribute, the routine verifies that the process ID that is found in each listed file is running. If you specify the MonitorProcesses attribute, the routine verifies that each listed process is running in the context you specify.

Use any combination among these attributes (MonitorProgram, PidFiles, or MonitorProcesses) to monitor the application.

If any of the processes that are specified in either PidFiles or MonitorProcesses is determined not to be running, the monitor returns OFFLINE. If the process terminates ungracefully, the monitor returns OFFLINE and failover occurs.

If the MonitorProgram attribute is specified to monitor the resource, the agent function performs the command:

```
su [-] user -c command_to_monitor_resource
```

imf_init Initializes the agent to interface with the asynchronous monitoring framework (AMF) kernel driver. This function runs when the agent starts up.

imf_getnotification Gets notification about resource state changes. This function runs after the agent initializes with the AMF kernel driver. The agent continuously waits for notification and takes action on the resource upon notification.

imf_register Registers the resource entities, which the agent must monitor, with the AMF kernel driver. For example, the function registers the PID for online monitoring of a process. This function runs for each resource after the resource goes into steady state (online or offline). The Application agent uses IMF for the processes configured with PidFiles and the MonitorProcesses attribute.

Clean — Terminates processes specified in PidFiles or MonitorProcesses. Ensures that only those processes (that are specified in the MonitorProcesses attribute) running with the user ID specified in the User attribute are killed. If the CleanProgram is defined, the agent executes the CleanProgram.

To forcefully stop the resource, the agent function performs the command:

```
su [-] user -c command_to_monitor_resource
```

Note that the agent uses the `su -` option only when the attribute UseSUDash is enabled (1). The UseSUDash attribute is disabled (0) by default.

State definitions

ONLINE — Indicates that all processes that are specified in the PidFiles and the MonitorProcesses attribute are running and that the MonitorProgram returns ONLINE.

OFFLINE — Indicates that at least one process that is specified in the PidFiles attribute or MonitorProcesses is not running, or that the MonitorProgram returns OFFLINE.

UNKNOWN — Indicates an indeterminable application state or invalid configuration.

FAULTED — Indicates that the process has terminated unexpectedly or MonitorProgram returns “offline” unexpectedly.

Attributes

Table 5-4 Required attributes

Required attribute	Description
StartProgram	The executable, created locally on each node, which starts the application. Specify the complete path of the executable. Applicable command line arguments follow the name of the executable and have spaces separating them. **Note:** Do not use the opening and closing ({ }) brace symbols in this string. Type and dimension: string-scalar Example: "/usr/sbin/sample_app start"
StopProgram	The executable, created locally on each node, which stops the application. Specify the complete path of the executable. Applicable command line arguments follow the name of the executable and have spaces separating them. **Note:** Do not use the opening and closing ({ }) brace symbols in this string. Type and dimension: string-scalar Example: "/usr/sbin/sample_app stop"
At least one of the following attributes: ■ MonitorProcesses ■ MonitorProgram ■ PidFiles	See “Optional attributes” on page 190.

Table 5-5 Optional attributes

Optional attribute	Description
CleanProgram	The executable, created locally on each node, which forcibly stops the application. Specify the complete path of the executable. Applicable command line arguments follow the name of the executable and have spaces separating them. Type and dimension: string-scalar Example: "/usr/sbin/sample_app stop"
MonitorProcesses	A list of processes that you want monitored and cleaned. Each process name is the name of an executable. Qualify the executable name with its complete path if the path starts the executable. The process name must be the full command line argument that the `ps -u user -eo pid,comm` command displays for the process. Type and dimension: string-vector
MonitorProgram	The executable, created locally on each node, which monitors the application. Specify the complete path of the executable. Applicable command line arguments follow the name of the executable and have spaces separating them. MonitorProgram can return the following states: OFFLINE value is 100; ONLINE values range from 101 to 110 (depending on the confidence level); 110 equals confidence level of 100%. Any other value = UNKNOWN. **Note:** Do not use the opening and closing ({ }) brace symbols in this string. Type and dimension: string-scalar Example: "/usr/sbin/sample_app_monitor all"

Table 5-5 Optional attributes

Optional attribute	Description
PidFiles	A list of files that contain the PID (process ID) of the processes that you want monitored and cleaned. These are application generated files. Each PID file contains one monitored PID. Specify the complete path of each PID file in the list. The process ID can change when the process restarts. If the application takes time to update the PID file, the agent's Monitor function may return an incorrect result. If incorrect results occur, increase the ToleranceLimit in the resource definition. Type and dimension: string-vector
User	The user name for running StartProgram, StopProgram, MonitorProgram, and CleanProgram. The processes that are specified in the MonitorProcesses list must run in the context of the specified user. Monitor checks the processes to make sure they run in this context. Type and dimension: string-scalar Default: root Example: user1
EnvFile	The environment file that should get sourced before running any of the StartProgram, StopProgram, MonitorProgram or CleanProgram. Type and dimension: string-scalar Default: "" Example: /home/username/envfile

Table 5-5 Optional attributes

Optional attribute	Description
UseSUDash	When the value of this attribute is 0, the agent performs an `su User` command before it executes the StartProgram, the StopProgram, the MonitorProgram, or the CleanProgram agent functions. When the value of this attribute is 1, the agent performs an `su - User` command before it executes the StartProgram, the StopProgram, the MonitorProgram or the CleanProgram agent functions. Type and dimension: boolean-scalar Default: 0 Example: 1

Resource type definition

```
type Application (
    static keylist SupportedActions = { "program.vfd", "user.vfd",
    "cksum.vfd", getcksum }
    static str ArgList[] = { User, StartProgram, StopProgram,
    CleanProgram, MonitorProgram, PidFiles, MonitorProcesses,
    EnvFile, UseSUDash}
    static int ContainerOpts{} = { RunInContainer=1, PassCInfo=0 }
    static str IMFRegList[] = { MonitorProcesses, User, PidFiles,
    MonitorProgram }
    str User = "root"
    str StartProgram
    str StopProgram
    str CleanProgram
    str MonitorProgram
    str PidFiles[]
    str MonitorProcesses[]
    str EnvFile
    boolean UseSUDash = 0
)
```

Application agent notes

Using Application agent with IMF

- Intelligent monitoring is supported for the Application agent only under specific configurations. The complete list of such configurations is provided in Table 5-6.

Table 5-6 Relation of monitoring attributes with IMF modes

MonitorProgram	MonitorProcesses	PidFiles	IMF Monitoring Mode
Not Configured	Not Configured	Not Configured	Not Applicable
Not Configured	Not Configured	Configured	Online Only
Not Configured	Configured	Not Configured	Online, Offline
Not Configured	Configured	Configured	Online, Offline
Configured	Not Configured	Not Configured	No IMF monitoring
Configured	Not Configured	Configured	No IMF monitoring
Configured	Configured	Not Configured	No IMF monitoring
Configured	Configured	Configured	No IMF monitoring

- When multiple processes are configured under the MonitorProcesses attribute and only some of them are running, offline registration with IMF will fail repeatedly until RegisterRetryLimit is reached. In such a scenario, IMF will not be able to determine when the resource goes ONLINE and the agent will monitor the resource in the traditional way.

Sample configurations

Configuration 1

In this example, you configure the executable sample_app as StartProgram and StopProgram, with start and stop specified as command line arguments respectively. Configure the agent to monitor two processes: a process that the app.pid specifies and the process sample_app.

```
Application samba_app (
    User = "root"
    StartProgram = "/usr/sbin/sample_app start"
    StopProgram = "/usr/sbin/sample_app stop"
    PidFiles = { "/var/lock/sample_app/app.pid" }
```

```
    MonitorProcesses = { "sample_app" }
)
```

Configuration 2

In this example, since no user is specified, it uses the root user. The executable sample_app starts and stops the application using start and stop as the command line arguments. The executable sample_app_monitor monitors the application and uses `all` as its command line argument. The agent also monitors the sample_app1 and sample_app2 processes.

```
Application samba_app2 (
    StartProgram = "/usr/sbin/sample_app start"
    StopProgram = "/usr/sbin/sample_app stop"
    CleanProgram = "/usr/sbin/sample_app force stop"
    MonitorProgram = "/usr/local/bin/sample_app_monitor all"
    MonitorProcesses = { "sample_app1", "sample_app2" }
)
```

Debug log levels

The Application agent uses the following debug log levels:

DBG_1, DBG_2, DBG_3, DBG_4, DBG_5

CoordPoint agent

Use the Coordination Point (CoordPoint) agent to monitor the registrations on the different coordination points on each node. You use this agent to provide server-based I/O fencing. The CoordPoint agent is a monitor-only agent that runs on each node within the client cluster. When you have configured a CP server as a coordination point, the CoordPoint agent performs the following tasks:

- Confirms that the CP server coordination point can communicate with the client cluster.
- Validates the node registrations in the CP server database using the `cpsadm` command.

In case the coordination point is a SCSI-3 based disk, the CoordPoint agent uses the `vxfenadm` command to confirm that the registered keys on the disk are intact. The Monitor agent function contains the monitoring functionality for SCSI-3 disks and CP servers.

If the agent detects an anomaly, the agent reports it to you so you can repair the coordination point. You may have to perform an online coordinator point replacement procedure if the problem is isolated to the keys registered.

Note: The CoordPoint agent that runs on a given client cluster node monitors the keys for coordination points visible to that node alone.

For important information about this agent, refer to:

"Notes for the CoordPoint agent" on page 197

Dependencies

No dependencies exist for the CoordPoint resource.

Agent functions

Monitor — Enables the CoordPoint agent to validate the node registrations in the coordination points and confirms that the coordination points are accessible.

CoordPoint resources are persistent, which means that they cannot be brought online or taken offline. They can only monitor the coordination point registrations. For this reason, the service group that contains the CoordPoint resource appears to be offline after a command such as `hastatus -sum`.

The CoordPoint agent also performs I/O fencing reporting activities.

See "CoordPoint agent I/O fencing reporting activities" on page 197.

State definitions

ONLINE	Indicates that the CoordPoint resource is working.
UNKNOWN	Indicates the agent cannot determine the coordination points resource's state. This state may be due to an incorrect configuration.
FAULTED	Indicates that the number of coordination points with missing keys (or registrations) has exceeded the value of the FaultTolerance attribute.

Attributes

Table 5-7 Required attributes

Required attribute	Description
FaultTolerance	The FaultTolerance attribute determines when the CoordPoint agent declares that the registrations on the coordination points are missing. If the number of coordination points with missing keys (or registrations) exceeds the value of the FaultTolerance attribute, then the agent reports FAULTED. Set the value of this attribute depending on your own configuration requirements. For example, if the FaultTolerance value is set to 1, then the CoordPoint agent reports FAULTED if it sees 2 or more number of coordinator points with missing keys (or registrations). Change the value of the FaultTolerance attribute either before the CoordPoint agent starts to monitor or while the CoordPoint agent is monitoring. If the attribute is set while the CoordPoint agent is monitoring, then the CoordPoint agent reads the new value in the next monitor cycle. To view the current FaultTolerance value, enter the following command: # **hares -display *coordpoint-res* -attribute FaultTolerance** Type and dimension: integer-scalar Default: "0"

Resource type definition

```
type CoordPoint (
    static str ArgList[] = { FaultTolerance }
    static int InfoInterval = 300
    static int OfflineMonitorInterval = 60
    static str Operations = None
    int FaultTolerance
)
```

Notes for the CoordPoint agent

CoordPoint agent I/O fencing reporting activities

The CoordPoint agent also performs the following I/O fencing reporting activities:

- Checks to determine if I/O fencing is running.
  If I/O fencing is not running, then the CoordPoint agent reports failure.
- Checks the mode of fencing operation. I/O fencing can operate in one of the following three modes:
  - SCSI-3 mode: If I/O fencing runs in SCSI-3 mode, then the CoordPoint agent continues to monitor.
  - Customized mode: If I/O fencing runs in Customized Fencing mode, then the CoordPoint agent continues to monitor.
  - Disabled mode: If I/O fencing runs in disabled mode, no action is required. The CoordPoint agent returns success.

AutoStartList attribute

AutoStartList is a service group attribute that needs to be populated with a system list. The VCS engine brings up the specified service group on the nodes in the list.

AutoStartList is not a required attribute for the service group that contains the CoordPoint resource. The CoordPoint resource is a persistent resource and when a service group is configured with this type of resource, it cannot be brought online.

Specifying the AutoStartList with a system list does not change the behavior of the service group. The service group will be reflected in OFFLINE status itself, irrespective of the AutoStartList attribute.

Sample configuration

In this example, the coordination point agent type resource is configured with the value of the FaultTolerance attribute set to 0. At this value setting, the CoordPoint agent reports FAULTED, when the agent determines that at least one coordination point has keys (or registrations) missing.

The following is an example service group (vxfen) extracted from a main.cf file:

```
group vxfen (
    SystemList = { sysA = 0, sysB = 1 }
    AutoFailOver = 0
    Parallel = 1
    AutoStartList = { sysA, sysB }
    )
        CoordPoint coordpoint (
            FaultTolerance=0
            )
    // resource dependency tree
    //
    //     group vxfen
    //     {
    //     CoordPoint coordpoint
    //     }
```

Debug log levels

The CoordPoint agent uses the following debug log levels:

DBG_1, DBG_10

Process agent

The Process agent starts, stops, and monitors a process that you specify. You can use the agent to make a process highly available.

This agent is Intelligent Monitoring Framework (IMF)-aware and uses asynchronous monitoring framework (AMF) kernel driver for IMF notification. For more information about IMF and intelligent resource monitoring, refer to the *Veritas Cluster Server Administrator's Guide*.

This agent is WPAR-aware. The ContainerOpts resource type attribute for this type has a default value of 1 for RunInContainer and a default value of 0 for PassCInfo. Symantec recommends that you do not change these values. Refer to the *Veritas Cluster Server Administrator's Guide* for more information.

Note that the AMF kernel driver also monitors the kernel processes if you have enabled intelligent monitoring for Process agent.

High availability fire drill

The high availability fire drill detects discrepancies between the VCS configuration and the underlying infrastructure on a node; discrepancies that might prevent a service group from going online on a specific node. For Process resources, the high availability fire drill checks for:

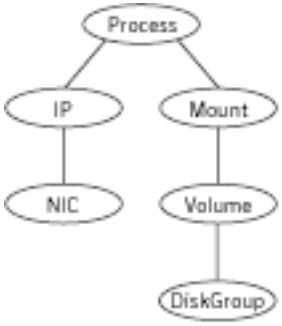
- The existence of a binary executable for the specified process (program.vfd)
- The existence of the same binary on all nodes (program.vfd)

For more information about using the high availability fire drill see the *Veritas Cluster Server Administrator's Guide*.

Dependencies

Depending on the context, this type of resource can depend on IP, IPMultiNIC, IPMultiNICB, and Mount resources.

Figure 5-3 Sample service group for a Process resource



Agent functions

Online	Starts the process with optional arguments.
Offline	Terminates the process with a `SIGTERM`. If the process does not terminate, a `SIGKILL` is sent.
Monitor	Checks to see if the process is running by scanning the process table for the name of the executable pathname and argument list.
imf_init	Initializes the agent to interface with the asynchronous monitoring framework (AMF) kernel driver. This function runs when the agent starts up.
imf_getnotification	Gets notification about resource state changes. This function runs after the agent initializes with the AMF kernel driver. The agent continuously waits for notification and takes action on the resource upon notification.
imf_register	Registers the resource entities, which the agent must monitor, with the AMF kernel driver. For example, the function registers the PID for online monitoring of a process. This function runs for each resource after the resource goes into steady state (online or offline).
Clean	Terminates all ongoing resource actions and takes the resource offline, forcibly when necessary.

State definitions

ONLINE	Indicates that the specified process is running in the specified user context. The agent only reports the process as online if the value configured for PathName attribute exactly matches the process listing from the ps output.
OFFLINE	Indicates that the specified process is not running in the specified user context.
FAULTED	Indicates that the process has terminated unexpectedly.
UNKNOWN	Indicates that the agent can not determine the state of the process.

Attributes

Table 5-8 Required attribute

Required attribute	Description
PathName	Complete pathname to access an executable program. This path includes the program name. If a script controls the process, the PathName defines the complete path to the shell. Type and dimension: string-scalar Example: "/usr/sbin/sendmail"

Table 5-9 Optional attributes

Optional attribute	Description
Arguments	Passes arguments to the process. If a script controls the process, the script is passed as an argument. Separate multiple arguments with a single space. A string cannot accommodate more than one space between arguments, nor allow for leading or trailing whitespace characters. This attribute must not exceed 80 characters. Type and dimension: string-scalar Example: "bd q1h"

Resource type definition

```
type Process (
    static keylist SupportedActions = { "program.vfd", getcksum }
    static str ArgList[] = { PathName, Arguments }
    static int ContainerOpts{} = { RunInContainer=1, PassCInfo=0 }
    str PathName
    str Arguments
)
```

Sample configurations

Configuration 1

```
Process  usr_lib_sendmail (
    PathName = "/usr/lib/sendmail"
    Arguments = "bd q1h"
    )
```

Configuration 2

```
include "types.cf"
cluster ProcessCluster (
.
.
.
group ProcessGroup (
    SystemList = { sysa = 0, sysb = 1 }
    AutoStartList = { sysa }
    )

    Process Process1 (
        PathName = "/usr/local/bin/myprog"
        Arguments = "arg1 arg2"
        )

    Process Process2 (
    PathName = "/bin/csh"
    Arguments = "/tmp/funscript/myscript"
    )

    // resource dependency tree
    //
    //    group ProcessGroup
    //    {
    //    Process Process1
    //    Process Process2
    //    }
```

Debug log levels

The Process agent uses the following debug log levels:

DBG_1, DBG_2, DBG_3, DBG_4, DBG_5

ProcessOnOnly agent

The ProcessOnOnly agent starts and monitors a process that you specify. You can use the agent to make a process highly available. This resource's Operation value is OnOnly.

VCS uses this agent internally to monitor security processes in a secure cluster.

Dependencies

No child dependencies exist for this resource.

Agent functions

Online	Starts the process with optional arguments.
Monitor	Checks to see if the process is alive by scanning the process table for the name of the executable pathname and argument list.
Clean	Terminates all ongoing resource actions and takes the resource offline, forcibly when necessary.

State definitions

ONLINE	Indicates that the specified process is running. The agent only reports the process as ONLINE if the value configured for PathName attribute exactly matches the process listing from the ps output.
FAULTED	Indicates that the process has unexpectedly terminated.
UNKNOWN	Indicates that the agent can not determine the state of the process.

Attributes

Table 5-10 Required attributes

Required attribute	Description
PathName	Defines complete pathname to access an executable program. This path includes the program name. If a process is controlled by a script, the PathName defines the complete path to the shell. The value configured for this attribute needs to match the process listing from the ps output for the agent to display as ONLINE. Type and dimension: string-scalar

Table 5-11 Optional attributes

Optional attribute	Description
Arguments	Passes arguments to the process. If a process is controlled by a script, the script is passed as an argument. Multiple arguments must be separated by a single space. A string cannot accommodate more than one space between arguments, nor allow for leading or trailing whitespace characters. Arguments must not exceed 80 characters (total). Type and dimension: string-scalar
IgnoreArgs	A flag that indicates whether monitor ignores the argument list. ■ If the value is 0, it checks the process pathname and argument list. ■ If the value is 1, it only checks for the executable pathname and ignores the rest of the argument list. Type and dimension: boolean-scalar Default: 0

Resource type definition

```
type ProcessOnOnly (
    static str ArgList[] = { IgnoreArgs, PathName, Arguments }
    static str Operations = OnOnly
    static int ContainerOpts{} = { RunInContainer=1, PassCInfo=0 }
    int IgnoreArgs
    str PathName
    str Arguments
)
```

Sample configurations

```
group VxSS (
    SystemList = { north = 0, south = 1 }
    Parallel = 1
    AutoStartList = { north, south }
    OnlineRetryLimit = 3
    OnlineRetryInterval = 120
    )

Phantom phantom_vxss (
    )

ProcessOnOnly vxatd (
    IgnoreArgs = 1
    PathName = "/opt/VRTSat/bin/vxatd"
    )
```

WPAR agent

The WPAR agent brings online, takes offline, and monitors workload partitions. You can use the agent to make WPARs highly available and to monitor them.

The ContainerOpts resource type attribute for this type has a default value of 0 for RunInContainer and a default value of 1 for PassCInfo. Symantec recommends that you do not change the values for these keys. Refer to the *Veritas Cluster Server Administrator's Guide* for more information.

Dependencies

No dependencies exist for the WPAR resource.

Figure 5-4 Sample service group that includes a WPAR resource

Agent functions

The value of the Operations attribute for this agent is OnOff.

Online	Brings a WPAR up and running.
Offline	Takes a WPAR down gracefully.
Monitor	Checks if the specified WPAR is up and running.
Clean	Another attempt to bring down a WPAR forcefully.

Attributes

Table 5-12 Optional attributes

Optional attribute	Description
ShutdownGracePeriod	Allows the root user to set the number of seconds before the shut down of a WPAR. Type and dimension: integer-scalar Default: 90 **Note:** Offline fails if the value of this attribute is 0 as the WPAR takes some time to shut down fully. Example: "10"
ResourceSet	A resource set is used to define a subset of processors in the system. If a resource set is specified for a workload partition, it can use the processors within the specified resource set only. The value of the ResourceSet attribute is the name of the resource set created using the `mkrset` command. If set, the agent configures the WPAR to use only the resource set specified by this attribute. Type and dimension: string-scalar Default: "" Example: ResourceSet = “myrset”
WorkLoad	Allows modification of resource control attributes shares_CPU and shares_memory. The key CPU is used to specify the number of processor shares that are available to the workload partition. The key MEM is used to specify the number of memory shares that are available to the workload partition. Type and dimension: integer-association Default: { } Example: { CPU = 50, MEM = 30 }

Resource type definition

```
type WPAR (
    static str ArgList[] = { ShutdownGracePeriod, ResourceSet,
    WorkLoad }
    static int ContainerOpts{} = { RunInContainer=0,
    PassCInfo=1 }
    int ShutdownGracePeriod = 90
    str ResourceSet
    int WorkLoad{}
)
```

For more information about configuring WPARs, refer to *Veritas Cluster Server Administrator's Guide*.

Debug log levels

The WPAR agent uses the following debug log levels:

DBG_1, DBG_5

MemCPUAllocator agent

Use the MemCPUAllocator agent to allocate CPU and memory to an IBM AIX dedicated partition. Set this resource's attribute values to specify the amount of CPU and memory that you want to allocate to a service group on a DLPAR. Configure this resource as a leaf node in the service group dependency tree.

For prerequisites and other important information about this agent, refer to:

"MemCPUAllocator agent notes" on page 213

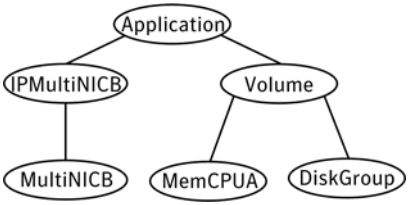
Platform

AIX

Dependencies

Set the MemCPUAllocator resource as a leaf node in a resource dependency tree. Select the amount of CPU and memory that you want the DLPAR to have before it comes online.

Figure 5-5 Sample service group that includes a MemCPUAllocator resource, where the MemCPUA resource represents the MemCPUAllocator resource



Agent functions

Online	The MemCPUAllocator agent dynamically allocates the required amount of memory and CPU to the DLPAR from the Hardware Management Console (HMC).
Offline	The agent deallocates the amount of memory and CPU it acquired during the online agent function. It then returns the resources back to the pool.

Monitor	Checks that the online agent function succeeded. If it succeeded, then the monitor agent function reports the resource state as ONLINE. If it did not succeed, then the monitor agent function reports the resource state as OFFLINE. If the agent is not able to allocate the required resources during the online agent function, the subsequent monitor reports OFFLINE and the resource faults. Because the resource is a leaf node, VCS engine stops bringing other resources online and marks the group as FAULTED. The VCS engine then tries to bring the group online on some other DLPAR. This check ensures that the agent can dynamically allocate the resources that the service group requires for the DLPAR.

Attributes

Table 5-13 Required attributes

Required attribute	Description
ManagedSystem	The name of the managed system that contains the partition. Type-dimension: string-scalar Example: mymachine
HMC	Name of the HMC The list of HMCs that control the managed systems. The agent tries to connect to any HMC on this list in the order that they are specified. Type-dimension: string-vector Example: HMC = { myhmc1, myhmc2 }

Table 5-14 Optional attributes

Optional attribute	Description
MemoryRequired	Amount of RAM (in MB) that you want to allocate. Type-dimension: string-scalar Default: 0 Example: 256
MemoryCritical	Specifies whether the memory allocation is critical. A value of 0 indicates that the online agent function should go ahead even when the required memory was not successfully allocated. Type-dimension: boolean-scalar Default: 0 Example: 1

Table 5-14 Optional attributes

Optional attribute	Description
CPURequired	The number of dedicated CPUs that you want to allocate. Type-dimension: string-scalar Example: 2
CPUCritical	Specifies whether the CPU allocation is critical. A value of 0 indicates that the online agent function should proceed even when the required CPU was not successfully allocated. Type-dimension: boolean-scalar Default: 0 Example: 1

Resource type definition

```
type MemCPUAllocator (
    static int NumThreads = 1
    static str ArgList[] = { ManagedSystem, HMC, MemoryRequired,
    MemoryCritical, CPUCritical, CPURequired }
    str ManagedSystem
    str HMC[]
    str MemoryRequired
    str CPURequired
    boolean CPUCritical = 0
    boolean MemoryCritical = 0
    temp boolean IsOnline = 0
)
```

MemCPUAllocator agent notes

The MemCPUAllocator agent has the following notes:

- See "Configuring password free SSH communication between VCS nodes and HMC" on page 214.
- See "Dynamic resource allocation scenarios" on page 214.
- See "Configuring MemCPUAllocator" on page 217.

Configuring password free SSH communication between VCS nodes and HMC

To use remote command operations on the HMC, you must have SSH installed on the DLPAR nodes in the VCS cluster. You must configure the HMC to allow password free SSH access from these partitions. Refer to the appropriate IBM AIX documentation for information.

To verify that you have password free SSH access

- From each DLPAR in the cluster, execute the following command to test if the password free access works.

```
Eagle> ssh -l hscroot hmc2.veritas.com
Last login:Thur Jun 16 22:46:51 2005 from 10.182.9.34
hscroot@hmc2:~>
```

  Once each node can connect to the HMC using SSH without a password, you can start to use the MemCPUAllocator agent.

Dynamic resource allocation scenarios

This section describes different examples of the resource allocation scenarios that the MemCPUAllocator agent can handle. For ease of explanation, consider only the memory resource in these examples. CPU resource implementation is similar.

Consider two DLPARs named Eagle and Vulture. These DLPARs are configured with the following minimum and maximum values memory values.

Table 5-15 The minimum and maximum memory for the DLPARs Eagle and Vulture

DLPAR	Minimum	Maximum
Eagle	512 MB	2 GB
Vulture	512 MB	2 GB

Two service groups SG1 and SG2 have the following resource requirements.

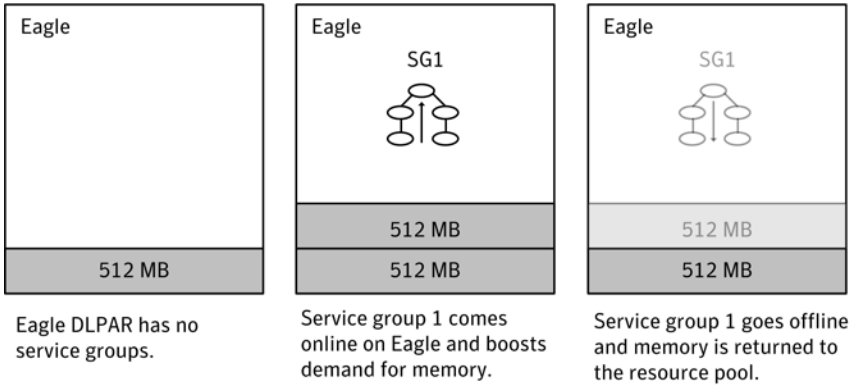
Table 5-16 The memory that is required for service group SG1 and SG2

Service group	Required memory
SG1	512 MB
SG2	512 MB

Scenario 1: A DLPAR node has minimum resources

Assume that the DLPARs start with the minimum values for memory. When SG1 is brought online on Eagle, the online agent function for the agent attempts to allocate 512 MB to Eagle from the free pool. The agent retains the minimum resources for the DLPAR's overhead operations and allocates resources for the service group in addition to the existing memory. For SG1 to come online the agent allocates an additional 512 MB to Eagle. After this allocation the total current memory for eagle is 1 GB. If SG1 goes offline, the agent deallocates the 512 MB that it allocated when the service group came online. This deallocation brings back the current memory of Eagle to 512 MB.

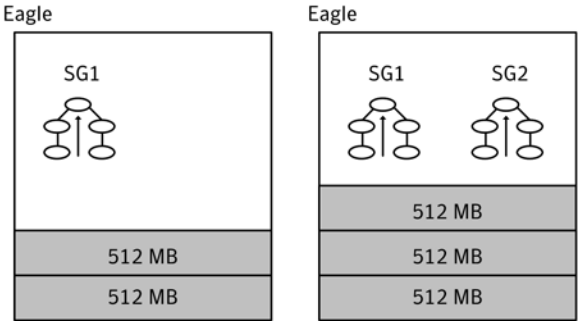
Figure 5-6 Bringing a service group online and taking it offline on a DLPAR



Scenario 2: Bringing another service group online

In this scenario, the Eagle DLPAR starts with 512 MB, and has SG1 online on it. It uses a total of 1 GB of memory. If SG2 is brought up on Eagle, the agent allocates an additional 512 MB of memory to Eagle. This reallocation brings the total memory to 1.5 GB.

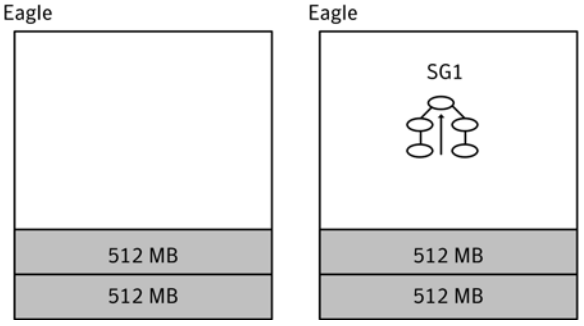
Figure 5-7 Bringing another service group online on a DLPAR



Scenario 3: DLPAR has required resources

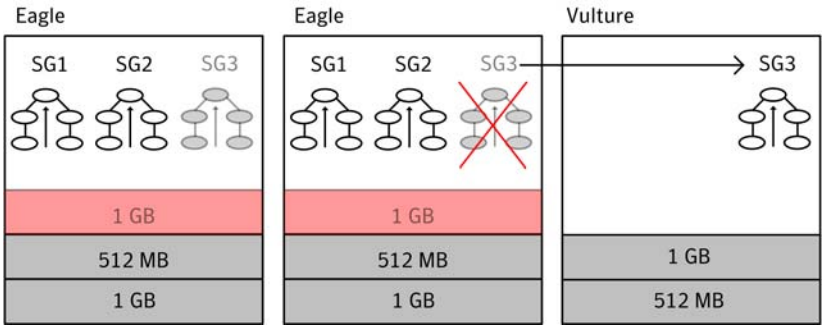
Instead of starting with 512 MB, Eagle starts with 1 GB of initial memory. Eagle has 512 MB more than its minimum amount. If SG1 is brought online on Eagle, the agent determines that Eagle has an extra 512 MB more than its minimum. No service groups use this extra 512 MB. The agent does not allocate any additional memory to Eagle. SG1 is brought online on Eagle and the current memory for Eagle stays 1 GB.

Figure 5-8 DLPAR Eagle starting with 1 GB of initial memory



Scenario 4: Cannot allocate required resources

Figure 5-9 Exceeding the maximum amount of memory on a DLPAR



Consider the stage in Scenario 2, where SG1 and SG2 are both online on Eagle, which brings its current memory to 1.5 GB. An additional service group SG3 enters the picture and requires 1 GB memory. SG3 tries to come up on Eagle. The agent determines that allocating 1 GB more memory to Eagle exceeds its maximum limit of 2 GB. The agent therefore does not allocate the memory and the online agent function fails, which leads to a resource fault. This resource fault makes the VCS engine stop the online of SG3 on Eagle and try it on Vulture. If Vulture starts with 512 MB and the agent allocates an additional 1 GB to Vulture, its current memory is 1.5 GB. SG3 can fail over and come online on Vulture.

Scenario 5: Service group failover

As in Scenario 2, SG1 and SG2 are both online on Eagle, which brings its current memory to 1.5 GB. Vulture has a current memory configuration of 512 MB. If you switch the service groups from Eagle to Vulture:

- The MemCPUAllocator agent's offline agent function deallocates 1 GB from Eagle (512 MB for SG1 and 512 MB for SG2).
- The VCS engine migrates SG1 and SG2 to Vulture and the agent's online agent function allocates 1 GB to Vulture. This allocation brings Vulture's memory to 1.5 GB.

Configuring MemCPUAllocator

Before you can use the MemCPUAllocator agent, you need to set up SSH access between the HMC and the DLPAR nodes. You must also make sure to configure the MemCPUAllocator resource as a leaf node in the service group's dependency tree in the main.cf file.

See Figure 5-5, “Sample service group that includes a MemCPUAllocator resource, where the MemCPUA resource represents the MemCPUAllocator resource,” on page 210.

Provide values to the MemCPUAllocator resource to specify the resource requirements for that service group. For example, if a service group needs 512 MB memory and two CPUs to start with, the MemCPUAllocator resource definition resembles:

```
MemCPUAllocator mymem (
    ManagedSystem @eagle = eagle-server
    ManagedSystem @vulture = vulture-server
    HMC = { testhmc }
    RequiredMemory = 512
    RequiredCPU = 2
    MemoryCritical = 1
    CPUCritical = 1
)
```

Debug log levels

The Mount agent uses the following debug log levels:

DBG_1, DBG_2

Chapter 6

Infrastructure and support agents

This chapter contains the following agents:

About the infrastructure and support agents

Use the infrastructure and support agents to monitor Veritas components and VCS objects.

NotifierMngr agent

Starts, stops, and monitors a notifier process, making it highly available. The notifier process manages the reception of messages from VCS and the delivery of those messages to SNMP consoles and SMTP servers. See the *Veritas Cluster Server Administrator's Guide* for a description of types of events that generate notification. See the `notifier`(1) manual page to configure notification from the command line.

You cannot dynamically change the attributes of the NotifierMngr agent using the `hares -modify` command. Changes made using this command are only effective after restarting the notifier.

Dependency

The NotifierMngr resource can depend on the NIC resource.

Agent functions

Online	Starts the notifier process with its required arguments.
Offline	VCS sends a `SIGABORT`. If the process does not exit within one second, VCS sends a `SIGKILL`.
Monitor	Monitors the notifier process.
Clean	Sends `SIGKILL`.

State definitions

ONLINE	Indicates that the Notifier process is running.
OFFLINE	Indicates that the Notifier process is not running.
UNKNOWN	Indicates that the user did not specify the required attribute for the resource.

Attributes

Table 6-1 Required attributes

Required attribute	Description
SnmpConsoles	Specifies the machine names of the SNMP managers and the severity level of the messages to be delivered. The severity levels of messages are Information, Warning, Error, and SevereError. Specifying a given severity level for messages generates delivery of all messages of equal or higher severity. **Note:** SnmpConsoles is a required attribute if SmtpServer is not specified; otherwise, SnmpConsoles is an optional attribute. Specify both SnmpConsoles and SmtpServer if desired. Type and dimension: string-association Example: "172.29.10.89" = Error, "172.29.10.56" = Information
SmtpServer	Specifies the machine name of the SMTP server. **Note:** SmtpServer is a required attribute if SnmpConsoles is not specified; otherwise, SmtpServer is an optional attribute. You can specify both SmtpServer and SnmpConsoles if desired. Type and dimension: string-scalar Example: "smtp.example.com"

Table 6-2 Optional attributes

Optional attribute	Description
MessagesQueue	Size of the VCS engine's message queue. Minimum value is 30. Type and dimension: integer-scalar Default: 30

Table 6-2 Optional attributes

Optional attribute	Description
NotifierListeningPort	Any valid, unused TCP/IP port number. Type and dimension: integer-scalar Default: 14144
NotifierSourceIP	If this attribute is populated, all the notifications sent from the notifier (SMTP and SNMP) will be sent from the interface having this IP address. **Note:** Make sure that the SourceIP given in this attribute is present in the /etc/hosts file or is DNS-resolvable. Type and dimension: string-scalar Example: "10.209.77.111"
SmtpFromPath	Set to a valid email address, if you want the notifier to use a custom email address in the FROM: field. Type and dimension: string-scalar Example: "usera@example.com"
SmtpRecipients	Specifies the email address where SMTP sends information and the severity level of the messages. The severity levels of messages are Information, Warning, Error, and SevereError. Specifying a given severity level for messages indicates that all messages of equal or higher severity are received. **Note:** SmtpRecipients is a required attribute if you specify SmtpServer. Type and dimension: string-association Example: "james@example.com" = SevereError, "admin@example.com" = Warning

Table 6-2 Optional attributes

Optional attribute	Description
SmtpReturnPath	Set to a valid email address, if you want the notifier to use a custom email address in the Return-Path: <> field. If the mail server specified in SmtpServer does not support SMTP VRFY command, then you need to set the SmtpVrfyOff to `1` in order for the SmtpReturnPath value to take effect. Type and dimension: string-scalar Example: "usera@example.com"
SmtpServerTimeout	This attribute represents the time in seconds notifier waits for a response from the mail server for the SMTP commands it has sent to the mail server. This value can be increased if you notice that the mail server is taking a longer duration to reply back to the SMTP commands sent by notifier. Type and dimension: integer-scalar Default: 10
SmtpServerVrfyOff	Set this value to `1` if your mail server does not support SMTP VRFY command. If you set this value to `1`, the notifier does not send a SMTP VRFY request to the mail server specified in SmtpServer attribute while sending emails. Type and dimension: boolean-scalar Default: 0
SnmpCommunity	Specifies the community ID for the SNMP manager. Type and dimension: string-scalar Default: public

Table 6-2 Optional attributes

Optional attribute	Description
SnmpdTrapPort	Port on the SNMP console machine where SNMP traps are sent. If you specify more than one SNMP console, all consoles use this value. Type and dimension: integer-scalar Default: 162
EngineListeningPort	Change this attribute if the VCS engine is listening on a port other than its default port. Type and dimension: integer-scalar Default: 14141

Resource type definition

```
type NotifierMngr (
    static int RestartLimit = 3
    static str ArgList[] = { EngineListeningPort, MessagesQueue,
    NotifierListeningPort, NotifierSourceIP, SnmpdTrapPort,
    SnmpCommunity, SnmpConsoles, SmtpServer, SmtpServerVrfyOff,
    SmtpServerTimeout, SmtpReturnPath, SmtpFromPath,
    SmtpRecipients }
    int EngineListeningPort = 14141
    int MessagesQueue = 30
    int NotifierListeningPort = 14144
    str NotifierSourceIP
    int SnmpdTrapPort = 162
    str SnmpCommunity = public
    str SnmpConsoles{}
    str SmtpServer
    boolean SmtpServerVrfyOff = 0
    int SmtpServerTimeout = 10
    str SmtpReturnPath
    str SmtpFromPath
    str SmtpRecipients{}
)
```

Sample configuration

In the following configuration, the NotifierMngr agent is configured to run with two resource groups: NicGrp and Grp1. NicGrp contains the NIC resource and a Phantom resource that enables VCS to determine the online and offline status of the group. See the Phantom agent for more information on verifying the status of groups that only contain OnOnly or Persistent resources such as the NIC resource. You must enable NicGrp to run as a parallel group on both systems.

Grp1 contains the NotifierMngr resource (ntfr) and a Proxy resource (nicproxy), configured for the NIC resource in the first group.

In this example, NotifierMngr has a dependency on the Proxy resource.

Note: Only one instance of the notifier process can run in a cluster. The process cannot run in a parallel group.

The NotifierMngr resource sets up notification for all events to the SNMP console `snmpserv`. In this example, only messages of SevereError level are sent to the SMTP server (`smtp.example.com`), and the recipient (`vcsadmin@example.com`).

Configuration

```
system north

system south

group  NicGrp (
    SystemList = { north, south }
    AutoStartList = { north }
    Parallel = 1
    )

    Phantom my_phantom (
    )

    NIC    NicGrp_en0 (
        Enabled = 1
        Device  = en0
        NetworkType = ether
        )

group Grp1 (
    SystemList = { north, south }
    AutoStartList = { north }
    )

    Proxy nicproxy(
```

```
    TargetResName = "NicGrp_en0"
    )

    NotifierMngr ntfr (
        SnmpConsoles = { snmpserv = Information }
        SmtpServer = "smtp.example.com"
        SmtpRecipients = { "vcsadmin@example.com" =
        SevereError }
    )

    ntfr requires nicproxy


    // resource dependency tree
    //
    //      group Grp1
    //      {
    //      NotifierMngr ntfr
    //              {
    //              Proxy nicproxy
    //              }
    //      }
```

IPv6 configuration

While the NotifierMngr resource can work without the NIC resource, Symantec recommends this dependency.

If the “en0” is a virtual device on AIX, then the NetworkHosts attribute is required, otherwise this resource takes an UNKNOWN state.

```
group ClusterService (
    SystemList = { sysA = 0, sysB = 1 }
    AutoStartList = { sysA, sysB }
    OnlineRetryLimit = 3
    OnlineRetryInterval = 120
    )

    NIC csgnic (
        Device = en0
        NetworkHosts = {"fe80::88c4:e0ff:fe00:c002"}
        )

    NotifierMngr ntfr (
        SnmpConsoles = { "3ffe:556::1000:5761" = SevereError }
        SmtpServer = "megami.veritas.com"
        SmtpRecipients = { "john_doe@symantec.com" =
        SevereError }
        )
        ntfr requires csgnic
```

Debug log levels

The NotifierMngr agent uses the following debug log levels:

DBG_1, DBG_2, DBG_3, DBG_5

Proxy agent

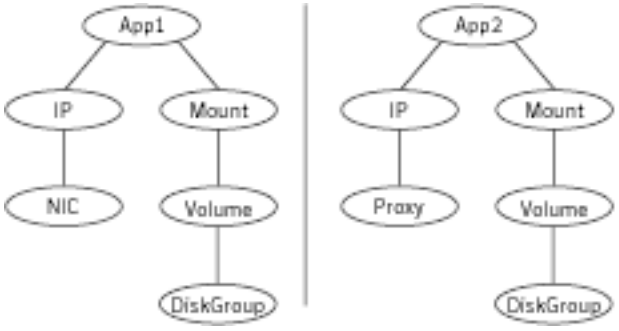
The Proxy agent mirrors the state of another resource on a local or remote system. It provides a means to specify and modify one resource and have its state reflected by its proxies. You can use the agent when you need to replicate the status of a resource.

A Proxy resource can only point to None or OnOnly type of resources, and can reside in a failover/parallel group. A target resource and its proxy cannot be in the same group.

Dependencies

No dependencies exist for the Proxy resource.

Figure 6-1 Sample service group that includes a Proxy resource



Agent functions

Monitor Determines status based on the target resource status.

Attributes

Table 6-3 Required attribute

Required attribute	Description
TargetResName	Name of the target resource that the Proxy resource mirrors. The target resource must be in a different resource group than the Proxy resource. Type and dimension: string-scalar Example: "tmp_VRTSvcs_file1"

Table 6-4 Optional attribute

Optional attribute	Description
TargetSysName	Mirrors the status of the TargetResName attribute on systems that the TargetSysName variable specifies. If this attribute is not specified, the Proxy resource assumes the system is local. Type and dimension: string-scalar Example: "sysa"

Resource type definition

```
type Proxy (
    static str ArgList[] = { TargetResName, TargetSysName,
    "TargetResName:Probed", "TargetResName:State" }
    static int OfflineMonitorInterval = 60
    static str Operations = None
    str TargetResName
    str TargetSysName
)
```

Sample configurations

Configuration 1

The proxy resource mirrors the state of the resource tmp_VRTSvcs_file1 on the local system.

```
Proxy proxy1 (
    TargetResName = "tmp_VRTSvcs_file1"
)
```

Configuration 2

The proxy resource mirrors the state of the resource tmp_VRTSvcs_file1 on sysa.

```
Proxy proxy1(
    TargetResName = "tmp_VRTSvcs_file1"
    TargetSysName = "sysa"
)
```

Configuration

The proxy resource mirrors the state of the resource mnic on the local system; note that target resource is in grp1, and the proxy is in grp2; a target resource and its proxy cannot be in the same group.

```
group grp1 (
    SystemList = { sysa, sysb }
    AutoStartList = { sysa }
    )

MultiNICA mnic (
        Device@sysa = { en0 = "10.128.8.42", en1 = "10.128.8.42" }
        Device@sysb = { en0 = "10.128.8.43", en1 = "10.128.8.43" }
        NetMask = "255.255.255.0"
        NameServerAddr = "10.130.8.1"
        Gateway = "10.128.1.1"
        Domain = "example.com"
        BroadcastAddr = "10.128.25.255"
        Options = "mtu m"
```

```
        )

        IPMultiNIC ip1 (
            Address = "166.98.14.78"
            NetMask = "255.255.255.0"
            MultiNICAResName = mnic
            Options = "mtu m"
            )
    ip1 requires mnic

group grp2 (
        SystemList = { sysa, sysb }
        AutoStartList = { sysa }
        )
        IPMultiNIC ip2 (
            Address = "166.98.14.79"
            NetMask = "255.255.255.0"
            MultiNICAResName = mnic
            Options = "mtu m"
            )
        Proxy proxy (
            TargetResName = mnic
            )
    ip2 requires proxy
```

Debug log levels

The Proxy agent uses the following debug log levels:

DBG_1, DBG_2

Phantom agent

The Phantom agent enables VCS to determine the state of parallel service groups that do not include OnOff resources.

Do not use the Phantom resource in failover service groups.

Note: Do not attempt manual online or offline operations on the Phantom resource at the resource level. Do not use `hares` commands on the Phantom resource at the resource level. Unpredictable behavior results when you try a manual online or offline procedure or an `hares` command on a Phantom resource. You can perform commands on the service group that contains the Phantom resource.

Dependencies

No dependencies exist for the Phantom resource.

Figure 6-2 Sample service group that includes a Phantom resource

Agent functions

Monitor Determines status based on the status of the service group.

Resource type definition

```
type Phantom (
    static str ArgList[] = { Dummy }
    str Dummy

)
```

Sample configurations

Configuration 1

```
Phantom boo (
)
```

Configuration 2

The following example shows a complete main.cf, in which the FileNone resource and the Phantom resource are in the same group.

```
include "types.cf"

cluster PhantomCluster

system sysa (
    )

system sysb (
    )

group phantomgroup (
    SystemList = { sysa = 0, sysb = 1 }
    AutoStartList = { sysa }
    Parallel = 1
    )

    FileNone my_file_none (
        PathName = "/tmp/file_none"
        )
    Phantom my_phantom (
        )

    // resource dependency tree
    //
    //    group maingroup
    //    {
    //    Phantom my_Phantom
    //    FileNone my_file_none
    //    }
```

RemoteGroup agent

The RemoteGroup agent establishes dependencies between applications that are configured on different VCS clusters. For example, you configure an Apache resource in a local cluster, and a MySQL resource in a remote cluster. In this example, the Apache resource depends on the MySQL resource. You can use the RemoteGroup agent to establish this dependency between these two resources.

With the RemoteGroup agent, you can monitor or manage a service group that exists in a remote cluster. Some points about configuring the RemoteGroup resource follow:

- For each remote service group that you want to monitor or manage, you must configure a corresponding RemoteGroup resource in the local cluster.
- Multiple RemoteGroup resources in a local cluster can manage corresponding multiple remote service groups in different remote clusters.
- You can include the RemoteGroup resource in any kind of resource or service group dependency tree.
- A combination of the state of the local service group and the state of the remote service group determines the state of the RemoteGroup resource.

Symantec supports the RemoteGroup agent when:

- When it points to a global group
  The RemoteGroup agent must then map the state of the global group in the local cluster.
- When it is configured inside a local parallel service group
  The RemoteGroup resources on all cluster nodes monitor the same remote service group unless its attributes are localized.
- When it is configured inside a local failover service group

For more information on the functionality of this agent see the *Veritas Cluster Server Administrator's Guide*.

Dependency

As a best practice, establish a RemoteGroup resource dependency on a NIC resource. Symantec recommends that the RemoteGroup resource not be by itself in a service group.

Agent functions

Online	Brings the remote service group online. See the "ControlMode" on page 237 for more information.
Offline	Takes the remote service group offline. See the "ControlMode" on page 237 for more information.
Monitor	Monitors the state of the remote service group. The true state of the remote service group is monitored only on the online node in the local cluster. See the "VCSSysName" on page 236.
Clean	If the RemoteGroup resource faults, the Clean function takes the remote service group offline. See the "ControlMode" on page 237 for more information.

State definitions

ONLINE	Indicates that the remote service group is in an ONLINE state. If the ReturnIntOffline attribute is not set to RemotePartial, then the remote service group is either in an ONLINE or PARTIAL state. See "ReturnIntOffline" on page 240.
OFFLINE	Indicates that the remote service group is in an OFFLINE or FAULTED state. The true state of the remote service group is monitored only on the online node in the local cluster. The RemoteGroup resource returns intentional offline if the attribute ReturnIntOffline is set to an appropriate value. See "ReturnIntOffline" on page 240.
FAULTED	Indicates that the RemoteGroup resource has unexpectedly gone offline.
UNKNOWN	Indicates that a problem exists either with the configuration or the ability of the RemoteGroup resource to determine the state of the remote service group.

Attributes

Table 6-5 Required attributes

Required attribute	Description
IpAddress	The IP address or DNS name of a node in the remote cluster. The IP address can be either physical or virtual. When configuring a virtual IP address of a remote cluster, do not configure the IP resource as a part of the remote service group. Type and dimension: string-scalar Examples: "www.example.com" or "11.183.12.214"
Port	This is a required attribute when the remote cluster listens on a port other than the default value of 14141. See "Port" on page 239.
GroupName	The name of the service group on the remote cluster that you want the RemoteGroup agent to monitor or manage. Type and dimension: string-scalar Example: "DBGrp"
VCSSysName	You must set this attribute to either the VCS system name or the ANY value. ■ ANY The RemoteGroup resource goes online if the remote service group is online on any node in the remote cluster. ■ *VCSSysName* Use the name of a VCS system in a remote cluster where you want the remote service group to be online when the RemoteGroup resource goes online. Use this to establish a one-to-one mapping between the nodes of the local and remote clusters. Type and dimension: string-scalar Example: "vcssys1" or "ANY"

Table 6-5 Required attributes

Required attribute	Description
ControlMode	Select only one of these values to determine the mode of operation of the RemoteGroup resource: MonitorOnly, OnlineOnly, or OnOff. ■ OnOff The RemoteGroup resource brings the remote service group online or takes it offline. When you set the VCSSysName attribute to ANY, the SysList attribute of the remote service group determines the node where the remote service group onlines. ■ MonitorOnly The RemoteGroup resource only monitors the state of the remote service group. The RemoteGroup resource cannot online or offline the remote service group. Make sure that you bring the remote service group online before you online the RemoteGroup resource. ■ OnlineOnly The RemoteGroup resource only brings the remote service group online. The RemoteGroup resource cannot take the remote service group offline. When you set the VCSSysName attribute to ANY, the SysList attribute of the remote service group determines the node where the remote service group onlines. Type and dimension: string-scalar

Table 6-5 Required attributes

Required attribute	Description
Username	This is the login user name for the remote cluster. When you set the ControlMode attribute to OnOff or OnlineOnly, the Username must have administrative privileges for the remote service group that you specify in the GroupName attribute. When you use the RemoteGroup Wizard to enter your username data, you need to enter your username and the domain name in separate fields. For a cluster that has the Symantec Product Authentication Service, you do not need to enter the domain name. For a secure remote cluster: ■ Local Unix user user@nodename—where the nodename is the name of the node that is specified in the IpAddress attribute. Do not set the DomainType attribute. ■ NIS or NIS+ user user@domainName—where domainName is the name of the NIS or NIS+ domain for the user. You must set the value of the DomainType attribute to either to nis or nisplus. Type and dimension: string-scalar Example: ■ For a cluster without the Symantec Product Authentication Service: "johnsmith" ■ For a secure remote cluster: "foobar@example.com"
Password	This is the password that corresponds to the user that you specify in the Username attribute. You must encrypt the password with the `vcsencrypt -agent` command. **Note:** Do not use the vcsencrypt utility when entering passwords from a configuration wizard or the Cluster Manager (Java Console). Type and dimension: string-scalar

Table 6-6 Optional attributes

Optional attribute	Description
DomainType	For a secure remote cluster only, enter the domain type information for the specified user. For users who have the domain type unixpwd, you do not have to set this attribute. Type: string-scalar Example: "nis", "nisplus"
BrokerIp	For a secure remote cluster only. If you need the RemoteGroup agent to communicate to a specific authentication broker, set the value of this attribute to the broker's IP address. Type: string-scalar Example: "128.11.295.51"
Port	The port where the remote engine listens for requests. This is an optional attribute, unless the remote cluster listens on a port other than the default value of 14141. Type and dimension: integer-scalar Default: 14141
OfflineWaitTime	The maximum expected time in seconds that the remote service group may take to offline. VCS calls the clean function for the RemoteGroup resource if the remote service group takes a longer time to offline than the time that you have specified for this attribute. Type and dimension: integer-scalar Default: 0

Table 6-6 Optional attributes

Optional attribute	Description
ReturnIntOffline	Select one of the following values for RemoteGroup to return IntentionalOffline: ■ RemotePartial—Indicates that the RemoteGroup resource returns an IntentionalOffline if the remote service group is in an ONLINE|PARTIAL state. ■ RemoteOffline—Indicates that the RemoteGroup resource returns an IntentionalOffline if the remote service group is in an OFFLINE state. ■ RemoteFaulted—Indicates that the RemoteGroup resource returns an IntentionalOffline if the remote service group is OFFLINE|FAULTED. You can use these values in combinations with each other. You must set the IntentionalOffline attribute of the RemoteGroup resource type to 1 for this attribute to work properly. For more information about this attribute, see the *Veritas Cluster Server Administrator's Guide*. Type and dimension: string-vector Default: ""
OfflineMonitoringNode	Defines the cluster node that performs the offline monitoring of the remote service group. This is an internal attribute. Do not modify.

Table 6-7 Type-level attributes

Type level attributes	Description
OnlineRetryLimit OnlineWaitLimit	In case of remote service groups that take a longer time to Online, Symantec recommends that you modify the default OnlineWaitLimit and OnlineRetryLimit attributes. See the *Veritas Cluster Server Administrator's Guide* for more information about these attributes.

Table 6-7 Type-level attributes

Type level attributes	Description
ToleranceLimit MonitorInterval	If you expect the RemoteGroup agent to tolerate sudden offlines of the remote service group, then modify the ToleranceLimit attribute. See the *Veritas Cluster Server Administrator's Guide* for more information about these attributes.
ExternalStateChange	If you want the local service group to go online or offline when the RemoteGroup resource goes online or offline outside VCS control, set the attribute ExternalStateChange appropriately. See the *Veritas Cluster Server Administrator's Guide* for more information about these attributes.

Resource type definition

```
type RemoteGroup (
    static int OnlineRetryLimit = 2
    static int ToleranceLimit = 1
    static boolean IntentionalOffline = 1
    static str ArgList[] = { IpAddress, Port, Username, Password,
    GroupName, VCSSysName, ControlMode, OfflineWaitTime,
    DomainType, BrokerIp, ReturnIntOffline }
    str IpAddress
    int Port = 14141
    str Username
    str Password
    str GroupName
    str VCSSysName
    str ControlMode
    int OfflineWaitTime
    str DomainType
    str BrokerIp
    str ReturnIntOffline[] = {}
    temp str OfflineMonitoringNode
)
```

Debug log levels

The RemoteGroup agent uses the following debug log levels:

DBG_1

Chapter 7

Testing agents

This chapter contains the following agents:

About the testing agents

Use the testing agents to provide high availability for program support resources. These resources are useful for testing service groups.

ElifNone agent

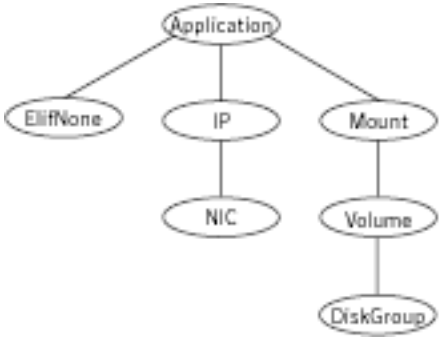
The ElifNone agent monitors a file. It checks for the file's absence.

You can use the ElifNone agent to test service group behavior. You can also use it as an impostor resource, where it takes the place of a resource for testing.

Dependencies

No dependencies exist for the ElifNone resource.

Figure 7-1 Sample service group that includes an ElifNone resource



Agent function

Monitor	Checks for the specified file. If it exists, the resource faults. If it does not exist, the agent reports as ONLINE.

State definitions

ONLINE	Indicates that the file specified in the PathName attribute does not exist.
FAULTED	Indicates that the file specified in the PathName attribute exists.
UNKNOWN	Indicates that the value of the PathName attribute does not contain a file name.

Attributes

Table 7-1 Required attribute

Required attribute	Description
PathName	Specifies the complete pathname. Starts with a slash (/) preceding the file name. Type and dimension: string-scalar Example: "/tmp/file01"

Resource type definition

```
type ElifNone (
    static str ArgList[] = { PathName }
    static int OfflineMonitorInterval = 60
    static str Operations = None
    str PathName
)
```

Sample configuration

```
ElifNone tmp_file01 (
    PathName = "/tmp/file01"
)
```

Debug log levels

The ElifNone agent uses the following debug log levels:

DBG_1, DBG_4, DBG_5

FileNone agent

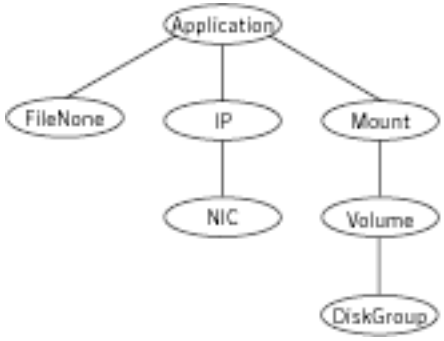
Monitors a file—checks for the file's existence.

You can use the FileNone agent to test service group behavior. You can also use it as an "impostor" resource, where it takes the place of a resource for testing.

Dependencies

No dependencies exist for the FileNone resource.

Figure 7-2 Sample service group that includes an FileNone resource



Agent functions

Monitor	Checks for the specified file. If it exists, the agent reports as ONLINE. If it does not exist, the resource faults.

State definitions

ONLINE	Indicates that the file specified in the PathName attribute exists.
FAULTED	Indicates that the file specified in the PathName attribute does not exist.
UNKNOWN	Indicates that the value of the PathName attribute does not contain a file name.

Attribute

Table 7-2 Required attribute

Required attribute	Description
PathName	Specifies the complete pathname. Starts with a slash (/) preceding the file name. Type and dimension: string-scalar Example: "/tmp/file01"

Resource type definition

```
type FileNone (
    static int AutoRestart = 1
    static int OfflineMonitorInterval = 60
    static str ArgList[] = { PathName }
    static str Operations = None
    str PathName
)
```

Sample configuration

```
FileNone tmp_file01 (
    PathName = "/tmp/file01"
)
```

Debug log levels

The FileNone agent uses the following debug log levels:

DBG_1, DBG_4, DBG_5

FileOnOff agent

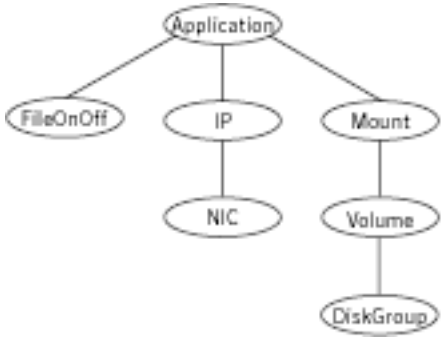
The FileOnOff agent creates, removes, and monitors files.

You can use this agent to test service group behavior. You can also use it as an "impostor" resource, where it takes the place of a resource for testing.

Dependencies

No dependencies exist for the FileOnOff resource.

Figure 7-3 Sample service group that includes a FileOnOff resource



Agent functions

Online	Creates an empty file with the specified name if the file does not already exist.
Offline	Removes the specified file.
Monitor	Checks for the specified file. If it exists, the agent reports as ONLINE. If it does not exist, the agent reports as OFFLINE.
Clean	Terminates all ongoing resource actions and takes the resource offline, forcibly when necessary.

State definitions

ONLINE	Indicates that the file specified in the PathName attribute exists.
OFFLINE	Indicates that the file specified in the PathName attribute does not exist.
FAULTED	Indicates that the file specified in the PathName attribute has been removed out of VCS control.
UNKNOWN	Indicates that the value of the PathName attribute does not contain a file name.

Attribute

Table 7-3 Required attribute

Required attribute	Description
PathName	Specifies the complete pathname. Starts with a slash (/) preceding the file name. Type and dimension: string-scalar Example: "/tmp/file01"

Resource type definition

```
type FileOnOff (
    static str ArgList[] = { PathName }
    str PathName
)
```

Sample configuration

```
FileOnOff tmp_file01 (
    PathName = "/tmp/file01"
)
```

Debug log levels

The FileOnOff agent uses the following debug log levels:

DBG_1, DBG_4, DBG_5

FileOnOnly agent

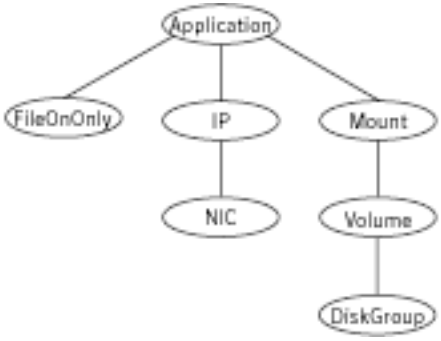
The FileOnOnly agent creates and monitors files.

You can use this agent to test service group behavior. You can also use it as an "impostor" resource, where it takes the place of a resource for testing.

Dependencies

No dependencies exist for the FileOnOnly resource.

Figure 7-4 Sample service group that includes a FileOnOnly resource



Agent functions

Online	Creates an empty file with the specified name, unless one already exists.
Monitor	Checks for the specified file. If it exists, the agent reports as ONLINE. If it does not exist, the resource faults.

State definitions

ONLINE	Indicates that the file specified in the PathName attribute exists.
OFFLINE	Indicates that the file specified in the PathName attribute does not exist and VCS has not attempted to bring the resource online.
FAULTED	Indicates that the file specified in the PathName attribute has been removed out of VCS control.

UNKNOWN Indicates that the value of the PathName attribute does not contain a file name.

Attribute

Table 7-4 Required attributes

Required attribute	Description
PathName	Specifies the complete pathname. Starts with a slash (/) preceding the file name. Type and dimension: string-scalar Example: "/tmp/file02"

Resource type definition

```
type FileOnOnly (
    static str ArgList[] = { PathName }
    static str Operations = OnOnly
    str PathName
)
```

Sample configuration

```
FileOnOnly tmp_file02 (
    PathName = "/tmp/file02"
)
```

Debug log levels

The FileOnOnly agent uses the following debug log levels:

DBG_1, DBG_4, DBG_5

Glossary

administrative IP address
The operating system controls these IP addresses and brings them up even before VCS brings applications online. Use them to access a specific system over the network for doing administrative tasks, for example: examining logs to troubleshoot issues, cleaning up temp files to free space, etc. Typically, you have one administrative IP address per node.

agent function
Agent functions start, stop, fault, forcibly stop, and monitor resources using scripts. Sometimes called an entry point.

base IP address
The administrative IP address of the system.

entry point
See agent function.

floating IP address
See virtual IP address.

logical IP address
Any IP address assigned to a NIC.

NIC bonding
Combining two or more NICs to form a single logical NIC, which creates a fatter pipe.

operation
All agents have scripts that turn the resource on and off. Operations determine the action that the agent passes to the resource. See None operation, OnOff operation, and OnOnly operation.

None operation
For example the NIC resource. Also called persistent resource, this resource is always on. This kind of resource has no online and offline scripts, and only monitors a resource.

OnOff operation
For example the IP and Share agents--in fact most agents are OnOff. This resource has online and offline scripts. Often this type of resource does not appear in the types file because by default when a resource does not have this resource type defined, it is OnOff.

OnOnly operation
For example the NFS, FileOnOnly resources. This kind of resource has an online script, but not an offline one.

plumb
Term for enabling an IP address—used across all platforms in this guide.

test IP address

IP addresses to help determine the state of a link by sending out a ping probe to another NIC (on another system.) Requires a return ping to complete the test. Test IP addresses can be the same as base IP addresses.

virtual IP address

IP addresses that can move from one NIC to another or from one node to another. VCS fails over these IP address with your application. Sometimes called a floating IP address.

Index

B

C

D

N

O

P

R

S

T

V

Z