

# Veritas™ Cluster Server Bundled Agents Reference Guide

HP-UX 11i v3

5.1 Service Pack 1

# 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.SP1.0

## Legal Notice

Copyright © 2011 Symantec Corporation. All rights reserved.

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

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

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

The Licensed Software and Documentation are deemed to be commercial computer software as defined in FAR 12.212 and subject to restricted rights as defined in FAR Section 52.227-19 "Commercial Computer Software - Restricted Rights" and DFARS 227.7202, "Rights in Commercial Computer Software or Commercial Computer Software Documentation", as applicable, and any successor regulations. Any use, modification, reproduction, release, performance, display or disclosure of the Licensed Software and Documentation by the U.S. Government shall be solely in accordance with the terms of this Agreement.

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](http://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](http://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/](http://www.symantec.com/business/support/)

## Customer service

Customer service information is available at the following URL:

[www.symantec.com/business/support/](http://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	<a href="mailto:customercare_apac@symantec.com">customercare_apac@symantec.com</a>
Europe, Middle-East, and Africa	<a href="mailto:semea@symantec.com">semea@symantec.com</a>
North America and Latin America	<a href="mailto:supportsolutions@symantec.com">supportsolutions@symantec.com</a>

## 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](mailto: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 1	Introduction	
	Resources and their attributes .....	19
	Modifying agents and their resources .....	20
	Attributes .....	20
	Enabling debug log messages .....	21
Chapter 2	Storage agents	
	About the storage agents .....	23
	DiskGroup agent .....	24
	Dependencies .....	24
	Agent functions .....	24
	State definitions .....	26
	Attributes .....	27
	Resource type definition .....	31
	DiskGroup agent notes .....	31
	High availability fire drill .....	31
	Using volume sets .....	32
	Setting the noautoimport flag for a disk group .....	32
	Sample configurations .....	32
	DiskGroup resource configuration .....	32
	DiskGroup, Volume, and Mount dependencies configuration .....	33
	Debug log levels .....	33
	DiskGroupSnap agent .....	34
	Dependencies .....	34
	Agent functions .....	35
	State definitions .....	35
	Attributes .....	36
	DiskGroupSnap agent notes .....	36
	Configuring the SystemZones attribute for the fire drill service group .....	37
	Configuring the firedrill service group .....	37
	Adding the ReuseMntPt attribute to the ArgList attribute for the Mount agent type .....	37
	Configuration considerations .....	38
	Agent limitations .....	39

Resource type definition .....	39
Sample configurations .....	40
Typical main.cf configuration .....	41
Oracle main.cf configuration .....	43
Debug log levels .....	46
Volume agent .....	47
Dependencies .....	47
Agent functions .....	47
State definitions .....	48
Attributes .....	49
Resource type definition .....	49
Sample configuration .....	50
Debug log levels .....	50
VolumeSet agent .....	51
Dependencies .....	51
Agent functions .....	51
State definitions .....	52
Attributes .....	52
Resource type definition .....	52
Sample configurations .....	53
A configured VolumeSet that is dependent on a DiskGroup resource .....	53
Agent notes .....	53
Inaccessible volumes prevent the VolumeSet agent from coming online .....	53
Debug log levels .....	53
LVMLogicalVolume agent .....	54
Dependencies .....	54
Agent functions .....	54
State definitions .....	55
Attributes .....	56
Resource type definition .....	56
Physical volumes associated with volume groups .....	57
Sample configurations .....	57
Configuration .....	57
Debug log levels .....	57
LVMVolumeGroup agent .....	58
Dependencies .....	58
Agent functions .....	58
State definitions .....	59
Attributes .....	59
Resource type definition .....	59
Sample configurations .....	59

Configuration 1 .....	59
Configuration 2: LVMVolumeGroup, LVMLogicalVolume, and Mount Dependencies .....	60
Debug log levels .....	60
LVMCombo agent .....	61
Dependencies .....	61
Agent functions .....	61
State definitions .....	62
Attributes .....	62
Resource type definition .....	63
Sample configurations .....	63
Sample 1 .....	63
Sample 2: LVMCombo and Mount Dependencies .....	63
Debug log levels .....	64
Mount agent .....	65
Dependencies .....	65
Agent functions .....	66
State definitions .....	68
Attributes .....	69
Resource type definition .....	75
Mount agent notes .....	76
High availability fire drill .....	76
VxFS file system lock .....	76
IMF usage notes .....	77
Support for loopback file system for HP-UX .....	78
Enabling second level monitoring for the Mount agent .....	79
Sample configurations .....	79
Configuration 1 .....	79
Configuration 2 .....	79
Configuration 3 .....	80
Configuration 4 .....	80
Debug log levels .....	80

## Chapter 3 Network agents

About the network agents .....	81
Agent comparisons .....	81
IP and NIC agents .....	81
IPMultiNIC and MultiNICA agents .....	81
IPMultiNICB and MultiNICB agents .....	82
802.1Q trunking .....	83
IP agent .....	84
High availability fire drill .....	84
Dependencies .....	84

Agent functions .....	85
State definitions .....	85
Attributes .....	86
Resource type definition .....	88
Sample configurations .....	88
Configuration 1 .....	88
NetMask in decimal (base 10) .....	88
NetMask in hexadecimal (base 16) .....	89
Debug log levels .....	89
NIC agent .....	90
High availability fire drill .....	90
Dependencies .....	90
Auto Port Aggregation (APA) support .....	91
Agent functions .....	91
State definitions .....	91
Attributes .....	92
Resource type definition .....	93
Sample configurations .....	94
Network Hosts .....	94
IPv6 configuration .....	94
Debug log levels .....	95
IPMultiNIC agent .....	96
Dependencies .....	96
Agent functions .....	96
State definitions .....	97
Attributes .....	98
Resource type definition .....	99
Debug log levels .....	100
MultiNICA agent .....	101
Dependencies .....	101
Agent function .....	102
State definitions .....	102
Attributes .....	103
Resource type definition .....	106
MultiNICA notes .....	107
General notes .....	107
Using RouteOptions .....	107
Sample configurations .....	108
MultiNICA and IPMultiNIC .....	108
Debug log levels .....	109
About the IPMultiNICB and MultiNICB agents .....	110
Checklist to ensure the proper operation of MultiNICB .....	110
IPMultiNICB agent .....	111

Dependencies .....	111
Requirements for IPMultiNICB .....	111
Agent functions .....	112
State definitions .....	112
Attributes .....	113
Resource type definition .....	115
Manually migrating a logical IP address .....	115
Sample configurations .....	115
Other sample configurations for IPMultiNICB and MultiNICB ..	115
Debug log levels .....	115
MultiNICB agent .....	116
About the MultiNICB agent .....	116
The haping utility .....	117
Auto Port Aggregation (APA) support .....	117
Dependencies .....	117
Agent functions .....	118
State definitions .....	118
Attributes .....	119
Resource type definition .....	123
Trigger script .....	123
IPMultiNICB and MultiNICB configuration .....	124
Debug log levels .....	124
DNS agent .....	125
Dependencies .....	125
Agent functions .....	126
State definitions .....	127
Attributes .....	128
Resource type definition .....	132
DNS agent notes .....	132
High availability fire drill .....	132
Monitor scenarios .....	133
Sample Web server configuration .....	133
Secure DNS update for BIND 9 .....	133
Setting up secure updates using TSIG keys for BIND 9 .....	133
Sample configurations .....	134
Basic IPv6 configuration .....	134
IPv6 CNAME sample configuration .....	135
IPv4 A sample configuration .....	135
Debug log levels .....	136

## Chapter 4

### File share agents

About the file service agents .....	137
NFS agent .....	138

Dependencies .....	138
Agent functions .....	138
State definitions .....	139
Attributes .....	139
Resource type definition .....	140
Sample configurations .....	140
Debug log levels .....	140
NFSRestart agent .....	141
Dependencies .....	141
Agent functions .....	142
State definitions .....	143
Attributes .....	144
Resource type definition .....	145
NFSRestart agent notes .....	145
About high availability fire drill .....	145
Providing a fully qualified host name .....	145
Sample configurations .....	146
Basic agent configurations .....	146
Debug log levels .....	147
Share agent .....	148
Dependencies .....	148
Agent functions .....	149
State definitions .....	149
Attributes .....	150
Resource type definition .....	150
Share agent notes .....	150
High availability fire drill .....	150
Sample configurations .....	151
Debug log levels .....	151
About the Samba agents .....	152
The Samba agents .....	152
Before using the Samba agents .....	152
Supported versions .....	153
Notes for configuring the Samba agents .....	153
Configuring multiple SambaServer resources .....	153
Configuring Samba for non-standard configuration files or non- standard lock directories .....	153
SambaServer agent .....	154
Dependencies .....	154
Agent functions .....	154
State definitions .....	155
Attributes .....	156
Resource type definitions .....	158

Sample configurations .....	158
Debug log levels .....	158
SambaShare agent .....	159
Dependencies .....	159
Agent functions .....	159
State definitions .....	159
Attributes .....	160
Resource type definition .....	160
Sample configuration .....	161
Debug log levels .....	161
NetBios agent .....	162
Dependencies .....	162
Agent functions .....	162
State definitions .....	163
Attributes .....	163
Resource type definition .....	165
Sample configuration .....	165
Debug log levels .....	166

## Chapter 5      Service and application agents

About the service and application agents .....	167
Apache Web server agent .....	168
Dependencies .....	168
Agent functions .....	169
State definitions .....	169
Attributes .....	170
Resource type definition .....	174
Apache Web server notes .....	174
Tasks to perform before you use the Apache Web server agent .....	175
About detecting application failure .....	175
About bringing an Apache Web server online outside of VCS control .....	176
About high Availability fire drill .....	176
Sample configurations .....	177
Application agent .....	178
High availability fire drill .....	178
Dependencies .....	178
Agent functions .....	180
State definitions .....	181
Attributes .....	182
Resource type definition .....	185
Application agent notes .....	186

Using Application agent with IMF .....	186
Sample configurations .....	186
Sample Configuration 1 .....	186
Sample Configuration 2 .....	187
Debug log levels .....	187
CoordPoint agent .....	188
Dependencies .....	188
Agent functions .....	188
State definitions .....	189
Attributes .....	189
Resource type definition .....	190
Notes for the CoordPoint agent .....	190
CoordPoint agent I/O fencing reporting activities .....	190
AutoStartList attribute .....	190
Sample configuration .....	191
Debug log levels .....	191
Process agent .....	192
High availability fire drill .....	192
Dependencies .....	192
Agent functions .....	193
State definitions .....	194
Attributes .....	195
Resource type definition .....	196
Sample configurations .....	196
Configuration 1 .....	196
Sample configuration 2 .....	197
Debug log levels .....	197
ProcessOnOnly agent .....	198
Dependencies .....	198
Agent functions .....	198
State definitions .....	198
Attributes .....	199
Resource type definition .....	200
Debug log levels .....	200
HPVirtualMachine agent .....	201
Limitations .....	201
Dependencies .....	201
Agent functions .....	201
State definitions .....	202
Attributes .....	203
Resource type definition .....	203
Sample configurations .....	204
Debug log levels .....	204

HPVSwitch agent .....	205
Dependencies .....	205
Agent functions .....	206
State definitions .....	206
Attributes .....	206
HPVSwitch agent notes .....	207
Agent limitation .....	207
Requirements .....	207
Resource type definition .....	207
Sample configurations .....	207
Creating an Integrity virtual machine service group .....	207
Service group with raw VxVM volume as the backing store .....	208
Service group with raw disk as the backing store .....	209
Service group with an online VM guest migration feature .....	210
Debug log levels .....	210

## Chapter 6 Infrastructure and support agents

About the infrastructure and support agents .....	211
NotifierMngr agent .....	212
Dependency .....	212
Agent functions .....	212
State definitions .....	212
Attributes .....	213
Resource type definition .....	216
Sample configuration .....	217
Configuration .....	217
Proxy agent .....	219
Dependencies .....	219
Agent functions .....	219
Attributes .....	220
Resource type definition .....	221
Sample configurations .....	221
Configuration 1 .....	221
Configuration 2 .....	221
Configuration .....	221
Debug log levels .....	222
Phantom agent .....	223
Dependencies .....	223
Agent functions .....	223
Resource type definition .....	223
Sample configurations .....	223
Configuration 1 .....	223
Configuration 2 .....	224

RemoteGroup agent .....	225
Dependency .....	225
Agent functions .....	226
State definitions .....	226
Attributes .....	227
Resource type definition .....	232
Debug log levels .....	232

## Chapter 7

### Testing agents

About the testing agents .....	233
ElifNone agent .....	234
Dependencies .....	234
Agent function .....	234
State definitions .....	234
Attributes .....	235
Resource type definition .....	235
Sample configuration .....	235
Debug log levels .....	235
FileNone agent .....	236
Dependencies .....	236
Agent functions .....	236
State definitions .....	236
Attribute .....	237
Resource type definition .....	237
Sample configuration .....	237
Debug log levels .....	237
FileOnOff agent .....	238
Dependencies .....	238
Agent functions .....	238
State definitions .....	239
Attribute .....	239
Resource type definition .....	239
Sample configuration .....	239
Debug log levels .....	239
FileOnOnly agent .....	240
Dependencies .....	240
Agent functions .....	240
State definitions .....	240
Attribute .....	241
Resource type definition .....	241
Sample configuration .....	241
Debug log levels .....	241

Glossary	243
Index	245



# 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	<p>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 (_).</p> <p>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 (\\).</p>
integer	<p>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.</p>

**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{}.

## 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 86.

# Storage agents

This chapter contains:

- [“DiskGroup agent”](#) on page 24
- [“DiskGroupSnap agent”](#) on page 34
- [“Volume agent”](#) on page 47
- [“VolumeSet agent”](#) on page 51
- [“LVMLogicalVolume agent”](#) on page 54
- [“LVMVolumeGroup agent”](#) on page 58
- [“LVMCombo agent”](#) on page 61
- [“Mount agent”](#) on page 65

## About the storage agents

Use storage agents to Monitor shared storage.

## DiskGroup agent

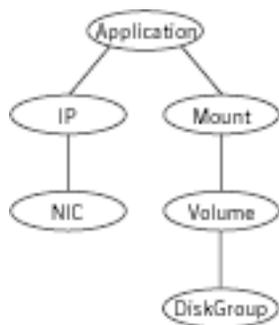
The DiskGroup agent brings online, takes offline, and monitors Veritas Volume Manager (VxVM) disk groups. This agent uses VxVM commands. You can use this agent to monitor or make disk groups highly available.

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.

## 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 <code>vxdg</code> command.
Offline	Deports the disk group using the <code>vxdg</code> command.
Monitor	<p>Determines if the disk group is online or offline using the <code>vxdg</code> command. The Monitor function changes the value of the VxVM <code>noautoimport</code> 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 <code>noautoimport</code> flag to on.</p> <pre># vxdg -g disk_group set autoimport=no</pre>
Clean	Terminates all ongoing resource actions and takes the resource offline—forcibly when necessary.

Info	<p>The DiskGroup info agent function gets information from the Volume Manager and displays the type and free size for the DiskGroup resource.</p> <p>Initiate the info agent function by setting the InfoInterval timing to a value greater than 0.</p> <p>In the following example, the info agent function executes every 60 seconds:</p> <pre># haconf -makerw # hatype -modify DiskGroup InfoInterval 60</pre> <p>The command to retrieve information about the DiskType and FreeSize of the DiskGroup resource is:</p> <pre># hares -value diskgroupres ResourceInfo</pre> <p>Output includes:</p> <pre>DiskType auto:cdsdisk FreeSize 12765712</pre>
Action	<p>Different action agent functions follow:</p> <ul style="list-style-type: none"><li>■ <code>license.vfd</code> Checks for valid Veritas Volume manager license—if one is not found use the <code>vxlicinst</code> utility to install a valid license key.</li><li>■ <code>disk.vfd</code> Checks if all disks in <code>diskgroup</code> 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.</li><li>■ <code>udid.vfd</code> 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.</li><li>■ <code>verifyplex.vfd</code> 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.</li><li>■ <code>volinuse</code> Checks if open volumes are in use or file systems on volumes that are mounted outside of VCS configuration.</li></ul> <p>See <a href="#">“High availability fire drill”</a> on page 31.</p>

## 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 Example: "diskgroup1"

**Table 2-2** Optional attributes

Optional attributes	Description
MonitorReservation	If the value 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	<p>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.</p> <p>If the value of this attribute is 1, and the disk group becomes disabled, the node panics.</p> <p>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.</p> <p><b>Note:</b> System administrators may want to set a high value for FaultOnMonitorTimeout to increase system tolerance.</p> <p>If the value of the attribute is 0, and the disk group becomes disabled, the following occurs:</p> <ul style="list-style-type: none"> <li>■ If the cluster has I/O fencing enabled, the DiskGroup resource is marked <code>FAULTED</code>. 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 <code>FAULTED</code> 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.</li> <li>■ If the cluster does not use I/O fencing, a message is logged and the resource is reported <code>ONLINE</code>. The resource is reported <code>ONLINE</code> so that it does not fail over, which ensures data integrity.</li> </ul> <p><b>Note:</b> The PanicSystemOnDGLoss attribute does not depend on the MonitorReservation attribute.</p> <p>Type and dimension: boolean-scalar</p> <p>Default: 0</p>

Table 2-2 Optional attributes

Optional attributes	Description
StartVolumes	<p>If value of this attribute is 1, the DiskGroup online function starts all volumes belonging to that disk group after importing the group.</p> <p><b>Note:</b> 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.</p> <p>Type and dimension: boolean-scalar Default: 1</p>
StopVolumes	<p>If value is 1, the DiskGroup offline function stops all volumes belonging to that disk group before it deports the disk group.</p> <p>Type and dimension: boolean-scalar Default: 1</p>
UmountVolumes	<p>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:</p> <ul style="list-style-type: none"><li>■ 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.</li><li>■ The agent attempts to forcefully unmount the file systems to close the volumes.</li></ul> <p>Type and dimension: boolean-scalar Default: 0</p>

**Table 2-2** Optional attributes

Optional attributes	Description
Reservation	<p>Determines if you want to enable SCSI-3 reservation. This attribute can have one of the following three values:</p> <ul style="list-style-type: none"> <li>■ 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.</li> <li>■ SCSI3—The disk group is imported with SCSI-3 reservation if the value of the cluster-level UseFence attribute is SCSI3.</li> <li>■ NONE—The disk group is imported without SCSI-3 reservation.</li> </ul> <p>Type and dimension: string-scalar            Default: ClusterDefault            Example: "SCSI3"</p>

**Table 2-3** Internal attributes

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

## Resource type definition

```
type DiskGroup (  
    static keylist SupportedActions = { "license.vfd", "disk.vfd",  
    "udid.vfd", "verifyplex.vfd", checkudid, numdisks, joindg,  
    splitdg, getvxvminfo, volinuse, campusplex }  
    static int NumThreads = 1  
    static int OnlineRetryLimit = 1  
    static str ArgList[] = { DiskGroup, StartVolumes, StopVolumes,  
    MonitorOnly, MonitorReservation, tempUseFence,  
    PanicSystemOnDGLoss, UmountVolumes, Reservation }  
    str DiskGroup  
    boolean StartVolumes = 1  
    boolean StopVolumes = 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”](#) on page 31
- [“Using volume sets”](#) on page 32
- [“Setting the noautoimport flag for a disk group”](#) on page 32

### 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.

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
```

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
```

## Sample configurations

### DiskGroup resource configuration

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

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

## DiskGroup, Volume, and Mount dependencies configuration

This sample configuration shows the DiskGroup, Volume, and Mount dependencies.

```
group sample_vxvm_group (
  SystemList = { System1, System2 }
  AutoStartList = { System1 }
)

Volume vres (
  Volume = voll
  DiskGroup = dg2
)

Mount mres (
  MountPoint = "/dir1"
  BlockDevice = "/dev/vx/dsk/dg2/voll"
  FSType = vxfs
  FsckOpt = "-y"
)

DiskGroup dres (
  DiskGroup = dg2
  StartVolumes = 0
  StopVolumes = 0
)

mres requires vres
vres requires dres
```

## 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*.

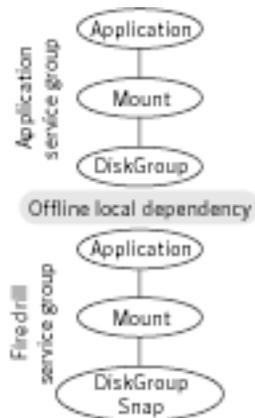
For important information about this agent, refer to:

“[DiskGroupSnap agent notes](#)” on page 36

## 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 fire drill 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	<p>The name of the DiskGroup resource from the application service group.</p> <p>Type-dimension: string-scalar</p> <p>Example: "dgres"</p>
FDSiteName	<p>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:</p> <ul style="list-style-type: none"><li>■ In the Gold configuration, a site has a dedicated set of fire drill disks. In <a href="#">Figure 2-4</a>, the disaster recovery site uses a Gold configuration.</li><li>■ In the Bronze configuration, a site uses its data disks as fire drill disks. In <a href="#">Figure 2-4</a>, the primary site uses a Bronze configuration.</li></ul> <p>Type and dimension: string-scalar</p> <p>Example:</p> <p>The value for the FDSiteName attribute for the configuration for <a href="#">Figure 2-4</a> is:</p> <pre>"FDSiteName@Node_A = pri" "FDSiteName@Node_B = pri" "FDSiteName@Node_C = dr_fd" "FDSiteName@Node_D = dr_fd"</pre>

## DiskGroupSnap agent notes

The DiskGroupSnap agent has the following notes:

- [“Configuring the SystemZones attribute for the fire drill service group”](#) on page 37
- [“Configuring the firedrill service group”](#) on page 37
- [“Adding the ReuseMntPt attribute to the ArgList attribute for the Mount agent type”](#) on page 37
- [“Configuration considerations”](#) on page 38
- [“Agent limitations”](#) on page 39

## 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
```

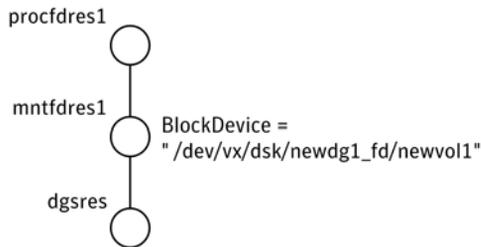
## Configuration considerations

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
```

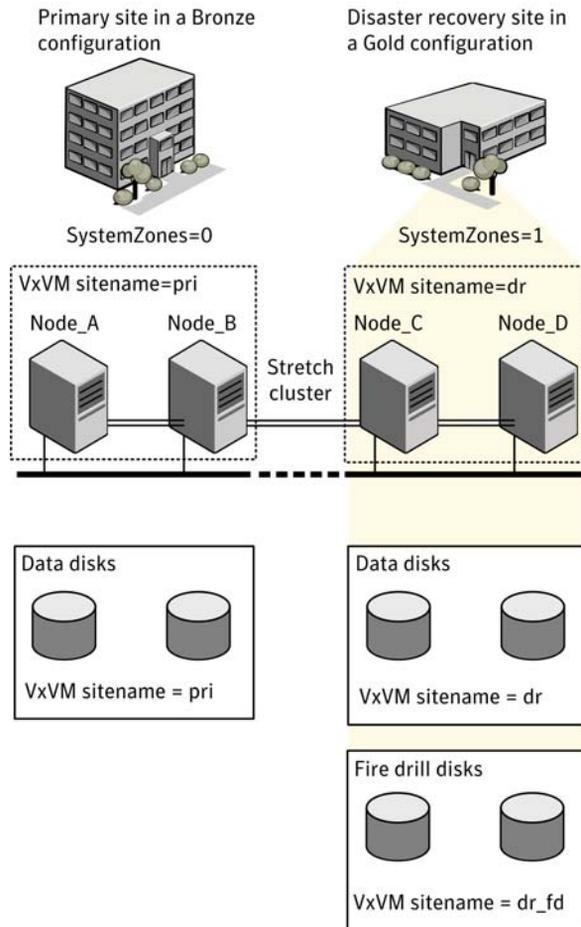
The Disaster Recovery site is in a Gold configuration as it has dedicated fire drill disks at the site. For the `FDSiteName` attribute, use the VxVM site tag given to the fire drill disks. For this configuration, the `FDSiteName` attribute values for the nodes at the Disaster recovery site follow:

```
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 procfres1 (
  PathName = "/usr/bin/ksh"
  Arguments = "/scrib.sh /dgsfs1"
)

Process procfres2 (
  PathName = "/usr/bin/ksh"
  Arguments = "/scrib.sh /dgsfs2"
)

requires group dgsg offline local
mntfdres1 requires dgsres
mntfdres2 requires dgsres
procfres1 requires mntfdres1
procfres2 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 proces1 (
  PathName = "/usr/bin/ksh"
  Arguments = "/scrib.sh /dgsfs1"
)

Process proces2 (
  PathName = "/usr/bin/ksh"
  Arguments = "/scrib.sh /dgsfs2"
)

mntres1 requires dgres
mntres2 requires dgres
proces1 requires mntres1
proces2 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 (
    FscckOpt = "-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 <code>vxrecover</code> command to start the volume.
Offline	Uses the <code>vxvol</code> 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 2-5** 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 2-6** 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

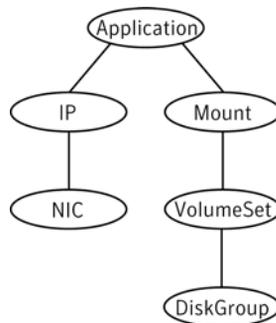
# VolumeSet agent

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 <code>vxrecover</code> command to start the volume set.
Offline	Uses the <code>vxvol</code> 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 2-7** 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

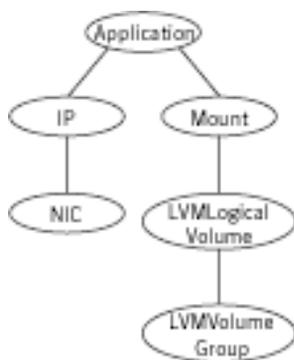
## LVMLogicalVolume agent

The LVMLogicalVolume agent brings online, takes offline, and monitors Logical Volume Manager (LVM) logical volumes. You can use this agent to make volume groups and logical volumes highly available and to monitor them.

### Dependencies

LVMLogicalVolume resources depend on LVMVolumeGroup resources.

**Figure 2-7** Sample service group that includes a LVMLogicalVolume resource



### Agent functions

Online	Activates the logical volume.
Offline	Deactivates the logical volume.
Monitor	Determines if the logical volume is accessible by performing read I/O on the raw logical volume.

## State definitions

ONLINE	Indicates that the Logical Volume is active.
OFFLINE	Indicates that the Logical Volume is not active.
UNKNOWN	Indicates that a problem exists either with the configuration or the ability to determine the status of the resource.

## Attributes

**Table 2-8** Required attributes

Required attribute	Description
LogicalVolume	Name of the logical volume. Type and dimension: string-scalar Example: "1vol1"
VolumeGroup	Name of the volume group containing the logical volume. Type and dimension: string-scalar Example: "vg1"

**Table 2-9** Optional attributes

Optional attribute	Description
VolumeIOTimeout	The time for which the agent should wait before it returns an OFFLINE state when IO to the volume hangs. Default: 15 Minimum value: 3 seconds Maximum value: No maximum value, but the higher the value the higher the failover time required.

## Resource type definition

```
type LVMLogicalVolume (  
    static int NumThreads = 1  
    static str ArgList[] = { LogicalVolume, VolumeGroup,  
        VolumeIOTimeout }  
    str LogicalVolume  
    str VolumeGroup  
    int VolumeIOTimeout = 15  
)
```

## Physical volumes associated with volume groups

For all the Physical Volumes (PV) that are associated with a volume group, set the timeout to a smaller value than specified in the VolumeIOTimeout attribute of the resource.

For example, if you specify an IOTimeout to equal 15 seconds, update the PV Timeout to a value that is less than 15 seconds.

Use the following command to change the timeout:

```
# pvchange -t time /dev/dsk/PV Used  
# pvchange -t time Physical Volume
```

For example:

```
# pvchange -t 10 /dev/dsk/c2t4d4
```

## Sample configurations

### Configuration

```
LVMLogicalVolume sharedg_lvoll (  
    LogicalVolume = lvoll  
    VolumeGroup = sharevg  
)
```

## Debug log levels

The LVMLogicalVolume agent uses the following debug log levels:

DBG\_1, DBG\_3

## LVMVolumeGroup agent

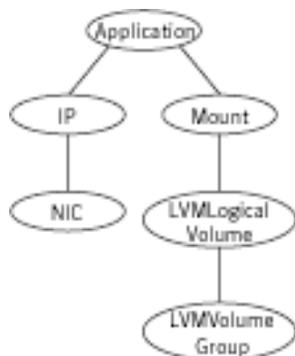
The LVMVolumeGroup agent activates, deactivates, and monitors LVM volume groups. You can use this agent to make volume groups and logical volumes highly available and to monitor them.

This agent supports Veritas Dynamic Multi-Pathing.

### Dependencies

The LVMVolumeGroup resource has no dependencies.

**Figure 2-8** Sample service group that includes a LVMVolumeGroup resource



### Agent functions

- |         |   |
|---------|---|
| Online  | Activates a volume group. While each system in the cluster must import the volume group, each system does not need to activate it.<br><br>This agent does not import volume groups because of the way LVM stores configuration information. Use the HP-UX SMH to import a volume group. |
| Offline | Deactivates a volume group with the <code>vgchange</code> command.  |
| Monitor | Determines whether the volume group is available.   |
| Clean   | Terminates all ongoing resource actions and takes the resource offline, forcibly when necessary.  |

## State definitions

ONLINE	Indicates that the volume group is active.
OFFLINE	Indicates that the volume group is not active.
UNKNOWN	Indicates that a problem exists either with the configuration or the ability to determine the status of the resource.

## Attributes

**Table 2-10** Required attributes

Required attribute	Description
VolumeGroup	The name of the volume group that is configured with Logical Volume Manager. Type and dimension: string-scalar Example: "sharevg"

## Resource type definition

```
type LVMVolumeGroup (  
    static keylist SupportedActions = { volinuse }  
    static str ArgList[] = { VolumeGroup }  
    str VolumeGroup  
)
```

## Sample configurations

### Configuration 1

```
LVMVolumeGroup sharevg (  
    VolumeGroup = sharevg  
)
```

## Configuration 2: LVMVolumeGroup, LVMLogicalVolume, and Mount Dependencies

This sample configuration shows the LVMVolumeGroup, LVMLogicalVolume, and Mount dependencies:

```
group sample_lvm (  
  SystemList = { System1, System2 }  
  AutoStartList = { System1 }  
)  
  
  LVMLogicalVolume lvolres (  
    LogicalVolume = lvol2  
    VolumeGroup = vg01  
  )  
  
  LVMVolumeGroup lvgres (  
    VolumeGroup = vg01  
  )  
  
  Mount mres (  
    MountPoint = "/dir2"  
    BlockDevice = "/dev/vg01/lvol2"  
    FSType = vxfs  
    MountOpt = ro  
    FsckOpt = "-y"  
  )  
  
  mres requires lvolres  
  lvolres requires lvgres
```

## Debug log levels

The LVMVolumeGroup agent uses the following debug log levels:

DBG\_1

## LVMCombo agent

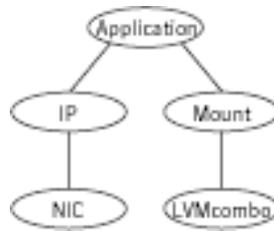
The LVMCombo agent controls the activation and deactivation of the logical volumes and the Logical Volume group. You can use this agent to make volume groups and logical volumes highly available.

This agent supports Veritas Dynamic Multi-Pathing.

### Dependencies

No dependencies exist for the LVMCombo resource.

**Figure 2-9** Sample service group that includes an LVMcombo resource



### Agent functions

Online	Activates the volume group and any of the logical volumes that are not available. While each system in the cluster must import the volume group, each system should not activate it.  This agent does not import volume groups because of the way LVM stores configuration information. Use the HP-UX SMH tool to import a volume group.
Offline	Deactivates the volume group, but does not deactivate the logical volumes. The logical volumes are automatically deactivated when the volume group is deactivated.
Monitor	If the volume group and all of the logical volumes are activated, the resource is online. Otherwise, the resource is reported offline.

---

**Note:** The monitor agent function does not perform any I/O on disk. If a disk that makes up a logical volume is powered off, the agent is not aware of this situation until LVM marks the logical volume unavailable. This situation may occur if the file system or the application using the logical volume attempts an I/O operation and fails. LVM can then set the logical volume as unavailable.

---

## State definitions

ONLINE	Indicates that the Volume Group and Logical Volumes are active.
OFFLINE	Indicates that the Volume Group and Logical Volumes are not active.
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
LogicalVolumes	List of logical volumes in a volume group. Type and dimension: string-vector Example: "lvol1" , "lvol2"
VolumeGroup	Name of a volume group. Type and dimension: string-scalar Example: "vg01"

**Table 2-12** Optional attributes

Optional Attribute	Description
VolumeIOTimeout	<p>The time for which the agent waits before it returns an OFFLINE state when I/Os to the volume hangs.</p> <p>Default: "15"</p> <p>Minimum value: 3 seconds</p> <p>Maximum value: No maximum value, but the higher the value the higher the failover time required.</p>

## Resource type definition

```

type LVMCombo (
    static keylist SupportedActions = { volinuse }
    static str ArgList[] = { VolumeGroup, LogicalVolumes,
        VolumeIOTimeout }
    str VolumeGroup
    str LogicalVolumes[]
    int VolumeIOTimeout = 15
)

```

## Sample configurations

### Sample 1

```

LVMCombo vg01 (
    VolumeGroup = vg01
    LogicalVolumes = { lvol1, lvol2 }
)

```

### Sample 2: LVMCombo and Mount Dependencies

This sample configuration shows the LVMCombo and Mount dependencies:

```

group sample_lvmcombo (
    SystemList = { System1, System2 }
    AutoStartList = { System1 }
)

LVMCombo lvmcbres (
    VolumeGroup = vg02
    LogicalVolumes = { lvol1 }
)

```

```
Mount mres (  
  MountPoint = "/dir2"  
  BlockDevice = "/dev/vg02/lvol1"  
  FSType = vxfs  
  MountOpt = ro  
  FsckOpt = "-y"  
)  
  
mres requires lvmcmbres
```

## Debug log levels

The LVMCombo agent uses the following debug log levels:

DBG\_1

## 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 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 is supported for VxFS and CFS mounts only.

---

The Mount agent supports the IPv6 protocol.

For important information about this agent, refer to:

“[Mount agent notes](#)” on page 76

## Dependencies

The Mount resource does not depend on any other resources.

**Figure 2-10** Sample service group that includes a Mount resource



## Agent functions

Online	<p>Mounts a block device on the directory. If the mount process fails for non-NFS mounts, the agent attempts to run the <code>fsck</code> command on the device before attempting to mount the file system again.</p> <p>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 <code>BlockDevice</code> attribute.</p>
Offline	<p>Unmounts the mounted file system gracefully.</p>
Monitor	<p>Determines if the file system is mounted.</p> <p>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.</p>
<code>imf_init</code>	<p>Initializes the agent to interface with the asynchronous monitoring framework (AMF) kernel driver. This function runs when the agent starts up.</p>
<code>imf_getnotification</code>	<p>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.</p>
<code>imf_register</code>	<p>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).</p>
Clean	<p>Unmounts the mounted file system forcefully.</p>

Info	<p>The Mount info agent function executes the command:</p> <pre>bdf mount_point</pre> <p>The output displays Mount resource information:</p> <pre>Size Used Avail Use%</pre> <p>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:</p> <pre>haconf -makerw hatype -modify Mount InfoInterval 60</pre> <p>The command to retrieve information about the Mount resource is:</p> <pre>hares -value mountres ResourceInfo</pre> <p>Output includes:</p> <pre>Size 2097152 Used 139484 Available 1835332 Used% 8%</pre>
Action	<ul style="list-style-type: none"><li>■ <b>chgmtlock</b> Resets the VxFS file system lock to a VCS-defined lock.</li><li>■ <b>mountpoint.vfd</b> 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 <code>mkdir</code> command.</li><li>■ <b>mounted.vfd</b> 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.</li><li>■ <b>vxfslic.vfd</b> Checks for valid Veritas File System (VxFS) licenses. If it fails, you need to update the license for VxFS.</li><li>■ <b>mountentry.vfd</b> Checks that the mount point is not listed in auto file system tables (for example, <code>/etc/fstab</code> ). If this action fails, you need to remove the mount point from auto file system tables.</li></ul>
attr_changed	<p>Unlocks the mounts when you change the value of the VxFSMountLock attribute from 1 (or 2) to 0 and vice-versa.</p>

## State definitions

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

## Attributes

**Table 2-13** Required attributes

Required attribute	Description
BlockDevice	<p>Block device for mount point.</p> <p>When you specify the block device to mount, enclose IPv6 addresses in square brackets. The <code>mount</code> 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.</p> <p>Type and dimension: string-scalar</p> <p>Examples:</p> <ul style="list-style-type: none"><li>■ VxVM "/dev/vx/dsk/myvcs_dg/myvol"</li><li>■ LVM "/dev/vg01/lvol01"</li><li>■ IPv4 "10.209.70.90:/dirname/anotherdir"</li><li>■ IPv6 "[fe80::1:2:3]:/dirname/anotherdir"</li></ul>
FsckOpt	<p>Mandatory for non-NFS mounts.</p> <p>Use this attribute to specify options for the <code>fsck</code> command. You must correctly set this attribute for local mounts. If the mount process fails, the <code>fsck</code> command is executed with the specified options before it attempts to remount the block device. Its value must include either <code>-y</code> or <code>-n</code>. Refer to the <code>fsck</code> manual page for more information.</p> <p>The <code>-y</code> argument enables the VxFS file systems to perform a log replay before a full <code>fsck</code> operation.</p> <p>For NFS mounts, the value of this attribute is not applicable and is ignored.</p> <p>Type and dimension: string-scalar</p> <p>VxFS example: <code>-y</code></p> <p><b>Note:</b> When you use the command line, add the <code>%</code> sign to escape <code>'</code>. For example: <code>hares -modify MntRes FsckOpt %y</code></p>

**Table 2-13** Required attributes

Required attribute	Description
FSType	Type of file system. Supports vxfs, hfs, lofs, or nfs. Type and dimension: string-scalar Example: "nfs"
MountPoint	Directory for mount point. Type and dimension: string-scalar Example: "/campus1"

Table 2-13 Required attributes

Required attribute	Description
VxFSMountLock	<p>This attribute is only applicable to vxfs file systems. This attribute controls a file system locking feature to prevent accidental unmounts.</p> <p>This attribute can take three values: 0, 1, or 2.</p> <p>VxFSMountLock=0</p> <p>The resource does not detect any changes to the lock when VCS reports that it is online after you set the value to zero.</p> <ul style="list-style-type: none"><li>■ If the mount point is initially locked with the mntlock="VCS", the monitor agent function unlocks it.</li><li>■ If the mount point is initially locked with a key that is not equal to "VCS", the agent logs a message once.</li><li>■ If the mount point is initially not locked, no action is performed.</li></ul> <p>VxFSMountLock=1</p> <p>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.</p> <ul style="list-style-type: none"><li>■ If the mount point is initially locked with the mntlock="VCS", no action is performed.</li><li>■ If the mount point is initially locked with a key that is not equal to "VCS", the agent logs a message once.</li><li>■ If the mount point is initially not locked, the monitor agent function locks it with the mntlock="VCS"</li></ul> <p>VxFSMountLock=2</p> <p>When the value of the VxFSMountLock is 2, the file system is locked and the agent monitors any change to mntlock.</p> <ul style="list-style-type: none"><li>■ If the mount point is locked with the mntlock="VCS", no action is performed.</li><li>■ If the mount point is initially locked with a key that is not equal to "VCS", the monitor agent function logs a message every monitor cycle.</li><li>■ If the mount point is not locked, the agent locks it with the mntlock="VCS".</li></ul> <p>Type and dimension: integer-scalar</p> <p>Default: 1</p>

**Table 2-14** Optional attributes

Optional attribute	Description
CkptUmount	<p>If the value of this attribute is 1, this attribute automatically unmounts VxFS Storage Checkpoints when the file system is unmounted.</p> <p>If the value of this attribute is 0, and Storage Checkpoints are mounted, then failover does not occur.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 1</p>
MountOpt	<p>Options for the <code>mount</code> command. Refer to the <code>mount</code> manual page for more information.</p> <p>Type and dimension: string-scalar</p> <p>Example: "rw"</p>
CreateMntPt	<p>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.</p> <p>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 of the mount point are less than 555, a warning message is logged.</p> <p>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.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 0</p>

Table 2-14 Optional attributes

Optional attribute	Description
MntPtGroup	<p>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.</p> <p>Type and dimension: string-scalar</p> <p>Example: "grp1"</p>
MntPtOwner	<p>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.</p> <p>Type and dimension: string-scalar</p> <p>Example: "usr1"</p>
MntPtPermission	<p>This attribute specifies the permissions of the mounted file system in 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.</p> <p>Type and dimension: string-scalar</p> <p>Example: "0755"</p>
AccessPermissionChk	<p>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.</p> <p>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.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 0</p>

**Table 2-14** Optional attributes

Optional attribute	Description
OptCheck	<p>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.</p> <p>If the value of this attribute is 0 (default), the mount options are not checked.</p> <p>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.</p> <p>If the value of the attribute is 1, the state of the resource is unaffected.</p> <p>If the value is 2, the state of the resource is set to offline.</p> <p>If the value is 3, state of the resource is set to unknown.</p> <p>Type and dimension: integer-scalar Default: 0</p>
RecursiveMnt	<p>If the value of this attribute is 1, VCS creates all the parent directories of the mount point if necessary.</p> <p>Type and dimension: boolean-scalar Default: 0</p>
SecondLevelMonitor	<p>This attribute has been deprecated.</p> <p>Instead of this attribute, use the LevelTwoMonitorFreq attribute. For more information, see <a href="#">“Enabling second level monitoring for the Mount agent”</a> on page 79.</p>
SecondLevelTimeout	<p>This attribute has been deprecated.</p>

**Table 2-14** Optional attributes

Optional attribute	Description
SnapUnmount	<p>If the value of this attribute is 1, this attribute automatically unmounts VxFS snapshots when the file system is unmounted.</p> <p>If the value of this attribute is 0 and snapshots are mounted, then failover does not occur.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 0</p>
ReuseMntPt	<p>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.</p> <p>To use this attribute, the cluster administrator needs to add this attribute to the arglist of the agent. Set the appropriate group and resource dependencies such that only one resource can come online on a system at a time.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 0</p>

## Resource type definition

```

type Mount (
    static keylist RegList = { VxFSMountLock }
    static keylist SupportedActions = { "mountpoint.vfd",
    "mounted.vfd", "mountentry.vfd", "vxfslic.vfd", chgmntlock }
    static str ArgList[] = { MountPoint, BlockDevice, FSType,
    MountOpt, FsckOpt, SnapUnmount, CkptUnmount, SecondLevelMonitor,
    SecondLevelTimeout, OptCheck, CreateMntPt, MntPtPermission,
    MntPtOwner, MntPtGroup, AccessPermissionChk, RecursiveMnt,
    VxFSMountLock, State }
    static str IMFRegList[] = { MountPoint, BlockDevice, FSType }
    str MountPoint
    str BlockDevice
    str FSType
    str MountOpt
    str FsckOpt
    int SnapUnmount
    int CkptUnmount = 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:

- [“High availability fire drill”](#) on page 76
- [“VxFS file system lock”](#) on page 76
- [“TMF usage notes”](#) on page 77
- [“Support for loopback file system for HP-UX”](#) on page 78
- [“Enabling second level monitoring for the Mount agent”](#) on page 79

### 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/fstab` 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 `vxumount` command with the option `mntunlock="key"`, for example:

```
# /opt/VRTS/bin/vxumount -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.

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.

## Support for loopback file system for HP-UX

The Mount agent provides loopback file system support. You can manage the loopback file system as a Mount resource. For loopback support, configure the `FSType` attribute to use a value of `lofs`.

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

```
Mount mntres (
    MountPoint = "/d"
    BlockDevice = "/dev/vx/dsk/test/testvol"
    FSType = vxfs
    FsckOpt = "-y"
    OptCheck = 1
)

Mount mntres1 (
    MountPoint = "/e"
    BlockDevice = "/d"
    FSType = lofs
    MountOpt = rw
    FsckOpt = "-y"
    OptCheck = 1
)

Volume volres (
    Volume = testvol
)

DiskGroup dgres (
    DiskGroup = test
)

mntres requires volres
mntres1 requires mntres
volres requires dgres
```

## 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

This sample configuration shows the LVMVolumeGroup, VMLogicalVolume, and Mount dependencies:

```
group sample_lvm (
    SystemList = { System1, System2 }
    AutoStartList = { System1 }
)
LVMLogicalVolume lvolres (
    LogicalVolume = lvol2
    VolumeGroup = vg01
)
LVMVolumeGroup lvgres (
    VolumeGroup = vg01
)
Mount mres (
    MountPoint = "/dir2"
    BlockDevice = "/dev/vg01/lvol2"
    FSType = vxfs
    MountOpt = ro
    FsckOpt = "-y"
)
mres requires lvolres
lvolres requires lvgres
```

### Configuration 2

In the following configuration, dg01vol is a volume in diskgroup dg01 created with VxVM. Mount resource mntres requires the volres volume resource and dgres diskgroup resource.

Give a complete configuration -

```
Mount mntres (  
    MountPoint = "/mnt"  
    BlockDevice = "/dev/vx/dsk/dg01/dg01vol"  
    FSType = vxfs  
    FsckOpt = "-y"  
)  
  
Volume volres (  
    Volume = dg01vol  
    DiskGroup = dg01  
)  
  
DiskGroup dgres (  
    DiskGroup = dg01  
)  
  
mntres requires volres  
volres requires dgres
```

### Configuration 3

In the following configuration, an `sysA` is the remote NFS server and `/tmp/source-dir` is the remote directory.

```
Mount mntres (  
    MountPoint = "/tmp/target-dir"  
    BlockDevice = "sysA:/tmp/source-dir"  
    FSType = nfs  
    FsckOpt = "-y"  
)
```

### Configuration 4

In the following configuration, an IPv6 address of the remote NFS server is used and `/tmp/source-dir` is the remote directory.

```
Mount mntres (  
    MountPoint = "/tmp/target-dir"  
    BlockDevice = "[2001::a:d1:48:c1]:/tmp/source-dir"  
    FSType = nfs  
)
```

## Debug log levels

The Mount agent uses the following debug log levels:

DBG\_1, DBG\_2, DBG\_3, DBG\_4, DBG\_5

# Network agents

This chapter contains the following:

- [“About the network agents”](#) on page 81
- [“IP agent”](#) on page 84
- [“NIC agent”](#) on page 90
- [“IPMultiNIC agent”](#) on page 96
- [“MultiNICA agent”](#) on page 101
- [“About the IPMultiNICB and MultiNICB agents”](#) on page 110
- [“IPMultiNICB agent”](#) on page 111
- [“MultiNICB agent”](#) on page 116
- [“DNS agent”](#) on page 125

## 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

#### 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 the System Management Homepage (SMH). The physical interface is connected to a 802.1Q trunked port on the switch.

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

For example, create a 802.1Q interface called lan9000 over a physical interface called lan0. Do not configure an IP address on lan0. You connect lan0 to a trunked port on the switch. The NIC and IP agents can then monitor lan9000 and the virtual IP address configured on lan9000. You must make sure that the IP addresses that are assigned to the interfaces of a particular VLAN are in the same subnet.

## 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 Auto-port Aggregation (APA).

## 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
- Checks for the validity of the configured PrefixLen attribute value for IPv6 configuration

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	Configures the IP address to the NIC. Checks if another system is using the IP address. Uses the <code>ifconfig</code> command to set the IP address on a unique alias on the interface.
Action	<p>The various functions of the action agent are as follows:</p> <ul style="list-style-type: none"><li>■ <code>route.vfd</code> Checks for the existence of a route to the IP from the specified NIC. Checks for the validity of the configured <code>PrefixLen</code> attribute value for IPv6 addresses.</li><li>■ <code>device.vfd</code> Checks for the existence of the interface configured in the <code>Device</code> attribute.</li></ul>
Offline	Brings down the IP address that is specified in the <code>Address</code> 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 attributes	Description
Address	<p>A virtual IP address, which is different from the base IP address, and which 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.</p> <p>Type and dimension: string-scalar</p> <p>Examples:</p> <p>IPv4: "192.203.47.61"</p> <p>IPv6: "2001::10"</p>
Device	<p>The name of the NIC device that is associated with the IP address. Contains the device name without an alias.</p> <p>Type and dimension: string-scalar</p> <p>Example: "lan0"</p>
PrefixLen	<p>Required to use the IPv6 protocol.</p> <p>See "<a href="#">PrefixLen</a>" on page 88.</p>

**Table 3-2** Optional attributes

Optional attributes	Description
ArpDelay	<p>The number of seconds to sleep between configuring an interface and sending out a broadcast to inform routers about this IP address.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 1</p>

Table 3-2 Optional attributes

Optional attributes	Description
IfconfigTwice	<p>Causes an IP address to be configured twice using an <code>ifconfig up-down-up</code> command sequence. Increases the probability of gratuitous ARP requests (generated by <code>ifconfig up</code>) to reach clients.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 0</p>
NetMask	<p>The netmask that is associated with the IP address of the resource. Specify the value of the netmask in decimal (base 10) or hexadecimal (base 16).</p> <p>Configure this attribute only if the IP address is an IPv4 address.</p> <p><b>Note:</b> Symantec recommends that you specify a netmask for each virtual interface.</p> <p>Type and dimension: string-scalar</p> <p>Example: "255.255.248.0"</p>
Options	<p>Options for the <code>ifconfig</code> command.</p> <p>Type and dimension: string-scalar</p> <p>Example: "broadcast 192.203.15.255"</p>
RouteOptions	<p>Specifies the routing options that are passed to the <code>route add</code> command when the agent configures an interface. The <code>RouteOptions</code> attribute value is generally formed like this: "<i>destination gateway metric</i>".</p> <p>For details about the <code>route</code> command, refer to the man page for your operating system.</p> <p>When the value of this string is null, the agent does not add routes.</p> <p>Type and dimension: string-scalar</p> <p>Example: "192.100.201.0 192.100.13.7"</p> <p>In this example, the agent executes the "<code>route add 192.100.201.0 192.100.13.7</code>" command when it configures an interface.</p>

**Table 3-2** Optional attributes

Optional attributes	Description
PrefixLen	<p>This is the prefix for the IPv6 address represented as the CIDR value.</p> <p>When you use the IPv6 protocol, you must configure values for this attribute and the corresponding NIC agent's Device and Protocol attributes.</p> <p>Type-dimension: integer-scalar</p> <p>Range: 1 - 128</p> <p>Example: 64</p>

## Resource type definition

```

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

```

## Sample configurations

### Configuration 1

```

IP ipres (
  Device = lan0
  Address = "192.203.47.61"
)

```

### NetMask in decimal (base 10)

```

IP ipres (
  Device = lan0
  Address = "192.203.47.61"
  NetMask = "255.255.248.0"
)

```

## NetMask in hexadecimal (base 16)

```
IP ipres (  
  Device = lan0  
  Address = "192.203.47.61"  
  NetMask = "0xfffff800"  
)
```

## Debug log levels

The IP agent uses the following debug log levels:

DBG\_1, DBG\_2, DBG\_3, DBG\_4, DBG\_5

## 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 Auto-port Aggregation (APA).

## 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.

## Auto Port Aggregation (APA) support

HP APA aggregates multiple network interfaces so that they appear as a single interface. For example you can combine lan0 and lan1 and call the combined interface lan9000. You then use the NIC agent to monitor the lan9000 interface. You use the IP agent to configure and monitor an IP address on the lan9000 interface. Note that you use the lan9000 interface configured through APA for the Device attribute.

The IP and NIC agents support APA use with VCS. APA is responsible for providing local adapter swapping, which is outside of VCS control.

## 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 <code>FAULTED</code> . 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.
---------	---

## 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	Name of the NIC that you want to monitor. Type and dimension: string-scalar Example: "lan0"
Protocol	Required to use the IPv6 protocol. See " <a href="#">Protocol</a> " on page 92.

**Table 3-4** Optional attributes

Optional attribute	Description
NetworkHosts	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 more than one network host is listed, the monitor returns ONLINE if at least one of the hosts is alive.  If an invalid network host address is specified or if there is mismatch in protocol of network host and Protocol attribute of the resource, the resource enters an UNKNOWN state.  Symantec recommends configuring at least one network host to report the correct NIC health status in case the network activity is low.  Type and dimension: string-vector Example: "166.96.15.22" , "166.97.1.2"
NetworkType	Type of network. VCS currently only supports Ethernet. Type and dimension: string-scalar Default: "ether"

**Table 3-4** Optional attributes

Optional attribute	Description
PingOptimize	<p>Allows or disallows broadcast pings to control the network traffic that the NIC agent generates.</p> <p>Use the PingOptimize attribute when you have not defined a value for the NetworkHosts attribute.</p> <p>A value of 1 optimizes broadcast pings—it disallows the NIC agent from sending broadcast ping requests.</p> <p>A value of 0 tells the agent to perform a broadcast ping during each monitor cycle and detects the inactive interface.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 1</p>
Protocol	<p>Specifies the type of IP protocol (IPv4 or IPv6) that you want to use with the agent.</p> <p>When you use the IPv6 protocol, you must configure values for this attribute, the Device attribute, and the corresponding IP agent's PrefixLen attribute.</p> <p>Type-dimension: string-scalar</p> <p>Default: IPv4</p> <p>Example: IPv6</p>

## Resource type definition

```

type NIC (
  static keylist SupportedActions = { "device.vfd" }
  static str ArgList[] = { Device, NetworkType, PingOptimize,
    NetworkHosts, Protocol }
  static int OfflineMonitorInterval = 60
  static str Operations = None
  str Device
  str NetworkType = ether
  int PingOptimize = 1
  str NetworkHosts[]
  str Protocol = IPv4
)

```

## Sample configurations

### Network Hosts

```
NIC groupx_lan0 (  
    Device = lan0  
    NetworkHosts = { "166.93.2.1", "166.99.1.2" }  
)
```

### 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, lan0).

```
group nic_group (  
    SystemList = { sysA = 0, sysB = 1 }  
)  
  
    NIC nic_resource (  
        Device@sysA = lan0  
        Device@sysB = lan1  
        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 = lan0  
        Device@sysB = lan1  
        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, DBG\_3, DBG\_4, DBG\_5

## IPMultiNIC agent

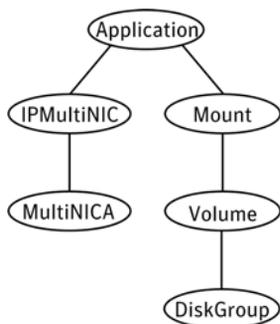
The IPMultiNIC agent manages the virtual IP address that is configured as an alias on one interface of a MultiNICA resource. If the interface faults, the agent works with the MultiNICA resource to fail over to a backup NIC. If multiple service groups have IPMultiNICs associated with the same MultiNICA 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.

**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.
Clean	Removes a virtual IP address from the interface where the virtual IP address is configured.
Open	Initializes the setup that the agent uses to start in a clean state.

Close Cleans up the setup that the agent uses.

## 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	<p>Virtual IP address assigned to the active NIC.</p> <p>Type and dimension: string-scalar</p> <p>Examples:</p> <ul style="list-style-type: none"> <li>■ IPv4: "10.128.10.14"</li> <li>■ IPv6: "2001:DB8::"</li> </ul>
MultiNICResName	<p>Name of the associated MultiNICA resource that determines the active NIC.</p> <p>Type and dimension: string-scalar</p> <p>Example: "mnic"</p>
PrefixLen	<p>Required to use the IPv6 protocol.</p> <p>See "<a href="#">PrefixLen</a>" on page 99.</p>

**Table 3-6** Optional attributes

Optional attribute	Description
IfconfigTwice	<p>Causes an IP address to be configured twice using an <code>ifconfig up-down-up</code> sequence. Increases the probability of gratuitous ARP requests (generated by <code>ifconfig up</code>) to reach clients.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 0</p>

**Table 3-6** Optional attributes

Optional attribute	Description
NetMask	<p>For the IPv4 protocol, the netmask that is associated with the IP address of the resource. Specify the value of the netmask in decimal (base 10) or hexadecimal (base 16).</p> <p>Configure this attribute only if the IP address is an IPv4 address.</p> <p><b>Note:</b> Symantec recommends that you specify a netmask for each virtual interface.</p> <p>Type and dimension: string-scalar</p> <p>Example: "255.255.255.0"</p>
Options	<p>Options for the <code>ifconfig</code> command.</p> <p>Type and dimension: string-scalar</p> <p>Example: "broadcast 192.203.15.255"</p>
PrefixLen	<p>Specifies the prefix for the IPv6 address represented as the CIDR value.</p> <p>When you use the IPv6 protocol, you must configure a value for this attribute.</p> <p>Type-dimension: integer-scalar</p> <p>Range: 1 - 128</p> <p>Example: 64</p>

## Resource type definition

```

type IPMultiNIC (
    static str ArgList[] = { "MultiNICResName:Device", Address,
        NetMask, "MultiNICResName:ArpDelay", Options,
        "MultiNICResName:Probed", MultiNICResName, IfconfigTwice,
        "MultiNICResName:Protocol", PrefixLen }
    str Address
    str NetMask
    str Options
    str MultiNICResName
    int IfconfigTwice
    static int MonitorTimeout = 120
    int PrefixLen
)

```

## Debug log levels

The IPMultiNIC agent uses the following debug log levels:

DBG\_1, DBG\_2, DBG\_3, DBG\_4, DBG\_5

## 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.

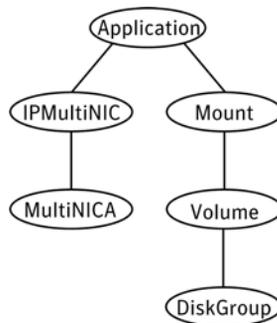
For important information on this agent, refer to:

See “[MultiNICA notes](#)” on page 107.

## 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.
---------	---

## 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
Device	List of interfaces and their base IP addresses. Type and dimension: string-association Example: { lan0 = "192.205.8.42", lan3 = "192.205.8.42" }
Protocol	Required to use the IPv6 protocol. See " <a href="#">Protocol</a> " on page 106.

**Table 3-8** Optional attributes

Optional attribute	Description
ArpDelay	Number of seconds to sleep between configuring an interface and sending out a broadcast to inform routers about the base IP address. Type and dimension: integer-scalar Default: 1

**Table 3-8** Optional attributes

Optional attribute	Description
HandshakeInterval	<p>Computes the maximum number of tries that the agent makes either to:</p> <ul style="list-style-type: none"> <li>■ ping a host (listed in the NetworkHosts attribute) when it fails over to a new NIC, or</li> <li>■ to ping the default broadcast address (depending on the attribute configured) when it fails over to a new NIC.</li> </ul> <p>To prevent spurious failovers, the agent must try to contact a host on the network several times before it marks a NIC as <code>FAULTED</code>. Increased values result in longer failover times, whether between the NICs or from system to system in the case of <code>FAULTED</code> NICs.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 20</p> <p>This value is the equivalent to two tries.</p>
IfconfigTwice	<p>Causes an IP address to be configured twice, using an <code>ifconfig up-down-up</code> sequence. Increases the probability of gratuitous ARP requests (caused by <code>ifconfig up</code>) to reach clients.</p> <p>Type and dimension: integer-scalar</p>
NetworkHosts	<p>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 can cause 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 an invalid network host address is specified or if there is a mismatch in protocol of the network host and Protocol attribute of resource, the resource enters an <code>UNKNOWN</code> state.</p> <p>Type and dimension: string-vector</p> <p>Example: "166.93.2.1", "166.97.1.2"</p>

Table 3-8 Optional attributes

Optional attribute	Description
NetMask	<p>Netmask for the base IP address. You can specify the value of NetMask in decimal (base 10) or hexadecimal (base 16).</p> <p><b>Note:</b> Symantec recommends that you specify a netmask for each virtual interface.</p> <p>Type and dimension: string-scalar</p> <p>Example: "255.255.255.0"</p>
Options	<p>The <code>ifconfig</code> options for the base IP address.</p> <p>Type and dimension: string-scalar</p> <p>Example: "broadcast 192.203.15.255"</p>
PingOptimize	<p>A value of 1 indicates that the agent not perform broadcast pings. A value of 0 performs a broadcast ping each monitor cycle and detects the inactive interface within the cycle.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 1</p>
RetestInterval	<p>Number of seconds to sleep between re-tests of a newly configured interface.</p> <p>A lower value results in faster local (interface-to-interface) failover.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 5</p>
RouteOptions	<p>String to add a route when configuring an interface. Use only when configuring the local host as the default gateway.</p> <p>The string contains destination gateway metric. No routes are added if this string is set to NULL.</p> <p>Type and dimension: string-scalar</p> <p>Example: "default 192.98.16.103 0"</p>

**Table 3-8** Optional attributes

Optional attribute	Description
PrefixLen	<p>Specifies the prefix for the IPv6 address represented as the CIDR value.</p> <p>When you use the IPv6 protocol, you must configure values for this attribute and the MultiNICA agent's Device and Protocol attributes.</p> <p>Type-dimension: integer-scalar</p> <p>Range: 1 - 128</p> <p>Example: 64</p>
Protocol	<p>Specifies the type of IP protocol (IPv4 or IPv6) that you want to use with the agent.</p> <p>When you use the IPv6 protocol, you must configure values for this attribute, the Device attribute, and the corresponding IPMultiNIC agent's PrefixLen attribute.</p> <p>Type-dimension: string-scalar</p> <p>Default: IPv4</p> <p>Example: IPv6</p>

## Resource type definition

```

type MultiNICA (
  static str ArgList[] = { Device, NetMask, ArpDelay,
    RetestInterval, Options, RouteOptions, PingOptimize,
    MonitorOnly, IfconfigTwice, HandshakeInterval,
    NetworkHosts, Protocol, PrefixLen }
  static int MonitorTimeout = 300
  static int OfflineMonitorInterval = 60
  static str Operations = None
  str Device{}
  str NetMask
  int ArpDelay = 1
  int RetestInterval = 5
  str Options
  str RouteOptions
  int PingOptimize = 1
  int IfconfigTwice
  int HandshakeInterval = 20
  str NetworkHosts[]
  str Protocol = IPv4

```

```
    int PrefixLen
  )
```

## MultiNICA notes

### General notes

- In a MultiNICA resource configuration, the link-local type of IPv6 addresses are not supported as the base address for device. The resource may enter a `FAULTED` state if the configuration contains a link-local address as a base address.
- 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.
- 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 HP-UX, for example, you have one active NIC, lan0 (10.128.2.5). You configure a second NIC, lan1, as the backup NIC to lan0. The agent does not fail over from lan0 to lan1 because all ping tests are redirected through lan0 on the same subnet. The redirect makes the MultiNICA monitor return an online status. Note that using ping `-i` does not enable the use of multiple active NICs.
- Before you start VCS, configure the primary NIC with the correct broadcast address and netmask.

### Using RouteOptions

The RouteOptions attribute is useful only when the default gateway is your own host.

For example, if the default gateway and lan0 are both set to 11.236.99.248, the output of the `netstat -rn` command from the routing table resembles:

Destination	Gateway	Flags	Refs	Interface	Pmtu
127.0.0.1	127.0.0.1	UH	0	lo0	4136
11.236.99.248	11.236.99.248	UH	0	lan0	4136
11.236.98.0	11.236.99.248	U	2	lan0	1500
127.0.0.0	127.0.0.1	U	0	lo0	0
default	11.236.99.248	UG	0	lan0	0

If the RouteOptions attribute is not set and lan0 fails, the MultiNICA agent migrates the base IP address to another NIC (such as lan1). The default route is no longer configured because it was associated with lan0. The display from the routing table resembles:

Destination	Gateway	Flags	Refs	Interface	Pmtu
127.0.0.1	127.0.0.1	UH	0	lo0	4136
11.236.99.161	11.236.99.161	UH	0	lan2	4136
11.236.98.0	11.236.99.161	U	2	lan2	1500

If the `RouteOptions` attribute defines the default route, the default route is reconfigured on the system. For example:

```
RouteOptions@sysa = "default 11.236.99.248 0"
RouteOptions@sysb = "default 11.236.99.249 0"
```

## Sample configurations

### MultiNICA and IPMultiNIC

In the following example, two systems, `sysa` and `sysb`, each have a pair of network interfaces, `lan0` and `lan3`. In this example, the two interfaces, `lan0` and `lan3`, 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 only the physical IP address to the backup NIC during a failure. The IPMultiNIC agent configures the logical IP addresses. The resources `ip1` and `ip2`, shown in the following example, have the `Address` attribute which contains the logical IP address. If a NIC fails on `sysa`, the physical IP address and the two logical IP addresses fails over from `lan0` to `lan3`. If `lan3` fails, the address fails back to `lan0` if `lan0` is reconnected.

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 second group 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. See the IPMultiNIC agent for more information.

```
group grp1 (
  SystemList = { sysa = 0, sysb = 1 }
  AutoStartList = { sysa }
)
MultiNICA mnic (
  Device@sysa = { lan0 = "192.205.8.42", lan3 = "192.205.8.42" }
  Device@sysb = { lan0 = "192.205.8.43", lan3 = "192.205.8.43" }
  NetMask = "255.255.255.0"
  ArpDelay = 5
  Options = "broadcast 192.203.15.255"
)
```

```
IPMultiNIC ip1 (  
  Address = "192.205.10.14"  
  NetMask = "255.255.255.0"  
  MultiNICResName = mnic  
  Options = "broadcast 192.203.15.255"  
)  
  
ip1 requires mnic  
  
group grp2 (  
  SystemList = { sysa = 0, sysb = 1 }  
  AutoStartList = { sysa }  
)  
  
IPMultiNIC ip2 (  
  Address = "192.205.9.4"  
  NetMask = "255.255.255.0"  
  MultiNICResName = mnic  
  Options = "broadcast 192.203.15.255"  
)  
  
Proxy proxy (  
  TargetResName = mnic  
)  
  
ip2 requires proxy
```

## Debug log levels

The MultiNICA agent uses the following debug log levels:

DBG\_1, 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.
- A MultiNICB resource controls all the interfaces on one IP subnet.
- 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.
- Reserve the base IP addresses, which the agent uses to test the link status, for use by the agent. These IP addresses do not get failed over.
- If you specify the NetworkHosts attribute, then that host must be on the same subnet as the other IP addresses for the MultiNICB resource.

## IPMultiNICB agent

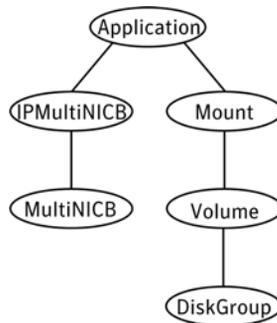
The IPMultiNICB agent works with the MultiNICB agent. The agent configures and manages virtual IP addresses (IP aliases) on an active network device that the MultiNICB resource specifies. When the MultiNICB agent reports a particular interface as failed, the IPMultiNICB agent moves the IP address to 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 Auto-port Aggregation (APA).

## 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.

## 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.

## State definitions

ONLINE	Indicates that the IP address is up on one of the working network interfaces of the MultiNICB resource. The IP address is specified in the Address attribute. The MultiNICB resource is specified in the BaseResName attribute.
OFFLINE	Indicates that the IP address is not up on any of the network interfaces of the MultiNICB resource. The IP address is specified in the Address attribute. The MultiNICB resource is specified in the BaseResName 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	The logical IP address that the IPMultiNICB resource must handle. Type and dimension: string-scalar Example: "192.205.10.15"
BaseResName	Name of MultiNICB resource from which the IPMultiNICB resource gets a list of working interfaces. The logical IP address is placed on the physical interfaces according to the device number information. Type and dimension: string-scalar Example: "gnic_n"
PrefixLen	Required to use the IPv6 protocol. See " <a href="#">PrefixLen</a> " on page 114.

**Table 3-10** Optional attributes

Optional attribute	Description
DeviceChoice	Indicates the preferred NIC where you want to bring the logical IP address online. Specify the device name or NIC alias as determined in the Device attribute of the MultiNICB resource. Type and dimension: string-scalar Default: 0 Examples: DeviceChoice = "lan0" DeviceChoice = "1"

**Table 3-10** Optional attributes

Optional attribute	Description
NetMask	<p>Netmask for the base IP address. You can specify the value of NetMask in decimal (base 10) or hexadecimal (base 16).</p> <p>Configure this attribute only if the IP address is an IPv4 address.</p> <p><b>Note:</b> Symantec recommends that you specify a netmask for each virtual interface.</p> <p>Type and dimension: string-scalar</p> <p>Example: "255.255.255.0"</p>
RouteOptions	<p>Specifies the routing options that are passed to the <code>route add</code> command when the agent configures an interface. The RouteOptions attribute value is generally formed like this: "<i>destination gateway metric</i>".</p> <p>For details about the <code>route</code> command, refer to the man page for your operating system.</p> <p>When the value of this string is null, the agent does not add routes.</p> <p>Type and dimension: string-scalar</p> <p>Example: "192.100.201.0 192.100.13.7"</p> <p>In this example, the agent executes the "<code>route add 192.100.201.0 192.100.13.7</code>" command when it configures an interface.</p>
Options	<p>Options for the <code>ifconfig</code> command.</p> <p>Type and dimension: string-scalar</p> <p>Example: "trailers"</p>
PrefixLen	<p>This is the prefix for the IPv6 address represented as the CIDR value.</p> <p>When you use the IPv6 protocol, you must configure values for this attribute and the corresponding MultiNICB agent's Device and Protocol attributes.</p> <p>Type-dimension: integer-scalar</p> <p>Range: 1 - 128</p> <p>Example: 64</p>

## Resource type definition

```
type IPMultiNICB (  
    static str ArgList[] = { BaseResName, Address, NetMask,  
        DeviceChoice, "BaseResName:Protocol", PrefixLen,  
        RouteOptions, Options }  
    str BaseResName  
    str Address  
    str NetMask  
    str DeviceChoice = 0  
    int PrefixLen  
    str RouteOptions  
    str Options  
)
```

## Manually migrating a logical IP address

Use the `haipswitch` command to migrate the logical IP address from one interface to another.

This command shows the status of the interfaces for the specified MultiNICB resource:

```
# haipswitch -s MultiNICB_resname
```

In the following example, the command checks that both the *from* and *to* interfaces are associated with the specified MultiNICB resource. The command also checks if the *to* interface works. If the interface does not work, the command aborts the operation. It then removes the IP address on the *from* logical interface and configures the IP address on the *to* logical interface. It finally erases any previous failover information that is created by MultiNICB for this logical IP address.

```
# haipswitch MultiNICB_resname IPMultiNICB_resname ip_address  
<netmask/prefix> from to [route_options]
```

## Sample configurations

### 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.

For the MultiNICB and IPMultiNICB agents, VCS supports Auto-port Aggregation (APA).

### About the MultiNICB agent

The agent sends packets to other hosts on the network to monitor the interfaces that it controls. It then checks the link status of the interfaces.

If a NIC goes down, the MultiNICB agent notifies the IPMultiNICB agent. The IPMultiNICB agent fails over the virtual IP addresses to a different NIC on the same system. When the original NIC comes up, the agents fail back the virtual IP address.

Each NIC must have its own unique and exclusive base IP address, which the agent uses as the test IP address.

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

MultiNICB uses the following criteria to determine if an interface works:

- **Interface status:** The interface status as reported by the driver of the interface (assuming that the driver supports this feature). This test is skipped if the attribute `IgnoreLinkStatus = 1`.
- **ICMP echo:** ICMP echo request packets are sent to one of the network hosts (if specified). Otherwise, the agent uses ICMP broadcast and caches the sender of the first reply as a network host. While the agent sends and receives ICMP packets, the IP layer is completely bypassed.

The MultiNICB agent writes the status of each interface to an export information file, which other agents (like IPMultiNICB) or commands (like `haipswitch`) can read.

#### Failover and failback

During an interface failure, the MultiNICB agent fails over all logical IP addresses to a working interface under the same resource. The agent remembers the first physical interface from which an IP address was failed over. This

physical interface becomes the “original” interface for the particular logical IP address. When the original interface is repaired, the logical IP address fails back to it.

## 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 this 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:

Examples for HPUX

Link test only on interface lan0:

```
haping -l lan0
```

Ping a remote host 10.10.10.10 from interface lan0:

```
haping -g 10.10.10.10 lan0
```

Ping a remote IPv6 host from interface lan0:

```
haping -p IPv6 -g fe80::1 lan0
```

## Auto Port Aggregation (APA) support

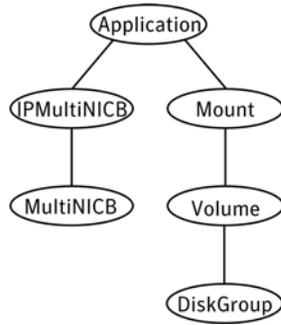
HP APA aggregates multiple network interfaces so that they appear as a single interface. For example you can combine lan0 and lan1 and call the combined interface lan9000. You then use the MultiNICB agent to monitor the lan9000 interface. You use the IPMultiNICB agent to configure and monitor an IP address on the lan9000 interface. Note that you use the lan9000 interface configured through APA for the Device attribute.

The IPMultiNICB and MultiNICB agents support APA use with VCS. APA is responsible for providing local adapter swapping, which is outside of VCS control.

## 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 attribute

Required attribute	Description
Device	<p>List of NICs that you want under MultiNICB control, and the aliases of those NICs. The IPMultiNICB agent uses the NIC aliases to configure IP addresses. The IPMultiNICB agent uses these interface aliases to determine the order of the interface on which to bring the IP addresses online.</p> <p>Type and dimension: string-association</p> <p>Example:</p> <p>Device = { "lan0" = 0, "lan1" = 2, "lan2" = 3 }</p> <p>In this example, the MultiNICB agent uses interfaces lan0, lan1, and lan2. The MultiNICB agent passes on the associated interface aliases 0, 2, and 3 to the IPMultiNICB agent.</p>
Protocol	<p>Required to use the IPv6 protocol.</p> <p>See <a href="#">“Protocol”</a> on page 122.</p>
NetworkHosts	<p>Required to use the IPv6 protocol.</p> <p>See <a href="#">“NetworkHosts”</a> on page 121.</p>

**Table 3-12** Optional attributes

Optional attribute	Description
DefaultRouter	<p>This attribute is the IP address of the default router on the subnet. If you specify this attribute, the agent removes the default route when the resource goes offline. The agent adds the route back when the group returns online. You must specify this attribute if multiple IP subnets exist on one host. If you do not specify the value, the packets cannot be routed properly when the subnet corresponding to the first default route goes down.</p> <p>Type and dimension: string-scalar</p> <p>Default: 0.0.0.0</p> <p>Example: "192.1.0.1"</p>

**Table 3-12** Optional attributes

Optional attribute	Description
Failback	<p>If the value of the attribute is 1, the virtual IP addresses are failed back to the original physical interface whenever possible. A value of 0 disables this behavior.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 0</p>
IgnoreLinkStatus	<p>If the value of the attribute is 1, the agent ignores the driver-reported interface status while testing the interfaces. If the value of the attribute is 0, the agent reports the interface status as down if the driver-reported interface status indicates the down state. Using interface status for link testing may considerably speed up failovers.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 1</p>
LinkTestRatio	<p>This attribute is the ratio of:</p> <ul style="list-style-type: none"> <li>■ The monitor cycles in which the agent tests the interfaces by sending packets, to</li> <li>■ The total monitor cycles</li> </ul> <p>At all other times, the agent tests the link by checking the "link-status" as reported by the device driver. Checking the "link-status" is a faster way to check the interfaces, but only detects cable disconnection failures.</p> <p>If the value of the attribute is 1, packets are sent during every monitor cycle.</p> <p>If the value of the attribute is 0, packets are never sent during a monitor cycle.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 1</p> <p>Example: "3"</p> <p>In this example, if the monitor agent function invokes in a numbered pattern such as 1, 2, 3, 4, 5, 6, ..., the actual packet send test is done at 3, 6, ... monitor agent functions. For LinkTestRatio=4, the packet send test is done at 4, 8, ... monitor agent functions.</p>

Table 3-12 Optional attributes

Optional attribute	Description
NetworkHosts	<p>List of host IP addresses on the IP subnet that are pinged to determine if the interfaces work. NetworkHosts only accepts IP addresses to avoid DNS lookup delays. The IP addresses must be directly present on the IP subnet of interfaces (the hosts must respond to ARP requests).</p> <p>If IP addresses are not provided, the hosts are automatically determined by sending a broadcast ping (unless the NoBroadcast attribute is set to 1). The first host to reply serves as the ping destination. 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.</p> <p>Type and dimension: string-vector Example: "192.1.0.1"</p>
NetworkTimeout	<p>Timeout for ARP and ICMP packets in milliseconds. MultiNICB waits for response to ICMP and ARP packets only during this time period.</p> <p>Assign NetworkTimeout a value in the order of tens of milliseconds (given the ICMP and ARP destinations are required to be on the local network). Increasing this value increases the time for failover.</p> <p>Type and dimension: integer-scalar Default: 100</p>
NoBroadcast	<p>If the value of the attribute is 1, NoBroadcast prevents MultiNICB from sending broadcast ICMP packets. (Note: MultiNICB can still send ARP requests.)</p> <p>If NetworkHosts are not specified and NoBroadcast is set to 1, the MultiNICB agent cannot function properly.</p> <p><b>Note:</b> Symantec does not recommend setting the value of NoBroadcast to 1.</p> <p>Type and dimension: integer-scalar Default: 0</p>

**Table 3-12** Optional attributes

Optional attribute	Description
OfflineTestRepeatCount	<p>Number of times the test is repeated if the interface status changes from UP to DOWN. For every repetition of the test, the next NetworkHost 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 also increases the response time.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 3</p>
OnlineTestRepeatCount	<p>Number of times the test is repeated if the interface status changes from DOWN to UP. This test helps to avoid oscillations in the status of the interface.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 3</p>
Protocol	<p>Specifies the type of IP protocol (IPv4 or IPv6) that you want to use with the agent.</p> <p>When you use the IPv6 protocol, you must configure values for this attribute, the Device attribute, and the corresponding IPMultiNICB agent's PrefixLen attribute.</p> <p>Type-dimension: string-scalar</p> <p>Default: IPv4</p> <p>Example: IPv6</p>

## Resource type definition

```
type MultiNICB (
    static int MonitorInterval = 10
    static int OfflineMonitorInterval = 60
    static str Operations = None
    static str ArgList[] = { Device, NetworkHosts, LinkTestRatio,
        IgnoreLinkStatus, NetworkTimeout, OnlineTestRepeatCount,
        OfflineTestRepeatCount, NoBroadcast, DefaultRouter, Failback,
        Protocol }
    str Device{}
    str NetworkHosts[]
    int LinkTestRatio = 1
    int IgnoreLinkStatus = 1
    int NetworkTimeout = 100
    int OnlineTestRepeatCount = 3
    int OfflineTestRepeatCount = 3
    int NoBroadcast
    str DefaultRouter
    int Failback
    str Protocol = IPv4
)
```

### 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:
  - lan0
- 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 `mnichb_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: Device lan0 status changed from down to up.

## IPMultiNICB and MultiNICB configuration

The following is an example VCS configuration.

```
include "types.cf"

cluster clus_north (
    UserNames = { admin = "cDRpdxPmHpzS." }
    Administrators = { admin }
    CounterInterval = 5
)

system north (
)

system south (
)

group g11 (
    SystemList = { north = 0, south = 1 }
    AutoStartList = { north, south }
)

IPMultiNICB ipmnicb (
    BaseResName = mnicb
    Address = "192.1.0.201"
    NetMask = "255.255.0.0"
    DeviceChoice = 1
)

MultiNICB mnicb (
    Device @north = { lan0 = 0, lan4 = 1 }
    Device @south = { lan0 = 0, lan4 = 1 }
    NetworkHosts = { "192.1.0.1" }
    DefaultRouter = "0.0.0.0"
)

ipmnicb requires mnicb
```

## Debug log levels

The MultiNICB agent uses the following debug log levels:

DBG\_1, DBG\_2, DBG\_3, DBG\_4, DBG\_5, DBG\_6

## 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.

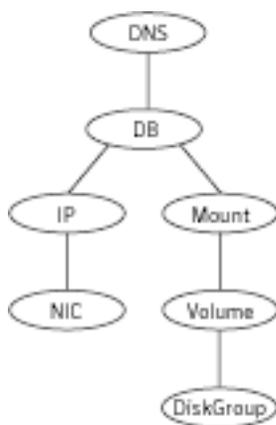
For important information about this agent, refer to:

“[DNS agent notes](#)” on page 132

## Dependencies

No dependencies exist for the DNS resource.

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



## Agent functions

Online	<p>Updates one or more name servers with the resource records.</p> <p>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.</p> <p>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.</p> <p>Finally the agent generates an Online lock file to indicate that the resource is online on the current system.</p>
Offline	<p>Removes the Online lock file.</p> <p>If attribute OffDelRR is true, offline removes all records that the ResRecord keys define.</p>
Monitor	<p>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.</p>
Clean	<p>Removes the Online lock file, if it exists.</p>
Open	<p>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.</p>
Action	<p>Different action agent functions follow:</p> <ul style="list-style-type: none"><li>■ keyfile.vfd This action entry point checks if the key file as specified in the TSIGKeyFile attribute exists either locally or on shared storage.</li><li>■ dig.vfd This action entry point checks if dig and nsupdate binaries exist and are executable.</li><li>■ master.vfd This action entry point checks if stealth masters are able to reply to SOA query for the configured domain.</li></ul>

## 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: <ul style="list-style-type: none"><li>■ The online lock does not exist.</li><li>■ None of the name servers can report all of the RRs' mappings.</li></ul>
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	<p>A string representing the DNS zone that the agent administers. The domain name can only contain alphanumeric symbols and the dash.</p> <p>Type and dimension: string-scalar</p> <p>Examples:</p> <ul style="list-style-type: none"><li>■ Forward mapping "demo.example.com"</li><li>■ IPv4 reverse mapping "2.168.192.in-addr.arpa"</li></ul>

**Table 3-13** Required attributes

Required attribute	Description
ResRecord	<p>You can use the ResRecord attribute alone, or you can use the Hostname and Alias attributes.</p> <p>ResRecord is an association of DNS resource record values. Each ResRecord attribute consists of two values: <i>DNS record key</i> = <i>DNS record data</i>. Note that the record key must be a unique value.</p> <p>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.</p> <p>Type and dimension: string-association</p> <p>Examples:</p> <ul style="list-style-type: none"> <li>■ For forward mapping, where the zone is demo.example.com: <ul style="list-style-type: none"> <li>- sles901 = "192.168.2.191"</li> <li>- ww2 = sles901</li> <li>- sles9ip6 = "2007::1:2:3:abc"</li> </ul> </li> <li>■ 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. <ul style="list-style-type: none"> <li>sle902 = "192.168.2.102 10.87.13.22"</li> </ul> A multi-home AAAA DNS record can be configured as follows: <ul style="list-style-type: none"> <li>sle902 = "1234::5678 1234::AABB:CCDD"</li> </ul> </li> <li>■ For reverse IPv4 address mapping, where the zone is 2.168.192.in-addr.arpa: <ul style="list-style-type: none"> <li>191 = "sles901.demo.example.com."</li> </ul> </li> <li>■ For reverse IPv6 address mapping, where the zone is 3.0.0.0.2.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.7.0.0.2.ip6.arpa: <ul style="list-style-type: none"> <li>cba = "sles9ip6.demo.example.com."</li> </ul> </li> </ul> <p>Use only partial host names. If you use a fully qualified domain name, append a period "." at the end of the name.</p> <p>For CNAME records, use:</p> <ul style="list-style-type: none"> <li>■ ResRecord = { www = mydesktop }</li> <li>or</li> <li>■ ResRecord = { www = "mydesktop.marketing.example.com." }</li> </ul> <p>Where the Domain attribute is "marketing.example.com"</p>

**Table 3-14** Required attributes

Required attribute	Description
ResRecord (cont.)	<p>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:</p> <ul style="list-style-type: none"> <li>■ PTR: if the Domain attribute ends with .arpa</li> <li>■ 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.</li> <li>■ AAAA: if the record data fields are in multiple sets of hexadecimal format, then this record is an IPv6 associated type AAAA record.</li> <li>■ CNAME: for any other valid record data.</li> </ul> <p><b>Note:</b> 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.</p>

**Table 3-15** Optional attributes

Optional attribute	Description
TTL	<p>A non-zero integer represents the “Time To Live” value, in seconds, for the DNS entries in the zone that you want to update.</p> <p>A lower value means more hits on your DNS server, while a higher value means more time for your clients to learn about changes.</p> <p>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.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 86400</p> <p>Example: 3600</p>

**Table 3-15** Optional attributes

Optional attribute	Description
StealthMasters	<p>The list of primary master name servers in the domain.</p> <p>This attribute is optional since the first name server is retrieved from the zone's SOA (Start of Authority) record.</p> <p>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.</p> <p>Type and dimension: string-vector</p> <p>Example: { "10.190.112.23" }</p>
TSIGKeyFile	<p>Required when you configure DNS for secure updates. Specifies the absolute path to the file containing the private TSIG (Transaction Signature) key.</p> <p>Type and dimension: string-scalar</p> <p>Example:</p> <pre>/var/tsig/example.com.+157+00000.private</pre>
CreatePTR	<p>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.</p> <p>Type and dimension: boolean-scalar</p> <p>Default: 0</p> <p>Example: 1</p>
OffDelRR	<p>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.</p> <p>The online agent function always adds records if they do not exist.</p> <p>Type and dimension: boolean-scalar</p> <p>Default: 0</p> <p>Example: 1</p>

## 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”](#) on page 132
- [“Monitor scenarios”](#) on page 133
- [“Sample Web server configuration”](#) on page 133
- [“Secure DNS update for BIND 9”](#) on page 133
- [“Setting up secure updates using TSIG keys for BIND 9”](#) on page 133

### 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

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.vcsd.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

# File share agents

This chapter contains the following:

- [“About the file service agents”](#) on page 137
- [“NFS agent”](#) on page 138
- [“NFSRestart agent”](#) on page 141
- [“Share agent”](#) on page 148
- [“About the Samba agents”](#) on page 152
- [“SambaServer agent”](#) on page 154
- [“SambaShare agent”](#) on page 159
- [“NetBios agent”](#) on page 162

## About the file service agents

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

## NFS agent

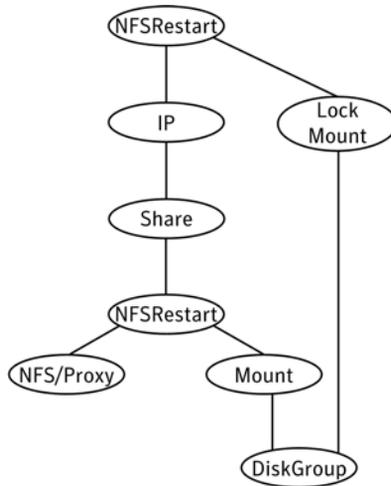
Starts and monitors the `nfsd` and `rpc.mountd` daemons required by all exported NFS file systems.

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.

## 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 <code>nfsd</code> and <code>rpc.mountd</code> daemons are running. If they are not running, the agent starts the daemons.
--------	---

Monitor	Monitors versions 2, 3, and 4 of the nfsd daemons, and versions 1 and 3 of the rpc.mountd daemon. Monitors TCP and UDP versions of the daemons by sending RPC (Remote Procedure Call) calls <code>clnt_create</code> and <code>clnt_call</code> to the RPC server. If the calls succeed, the resource is reported ONLINE.
Clean	Terminates and restarts the nfsd and rpc.mountd daemons.

## 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.
UNKNOWN	Unable to determine the status of the NFS daemons.

## Attributes

**Table 4-1** Optional attributes

Optional attribute	Description
LockFileTimeout	If the group goes offline, the agents waits for the specified time before restarting nfsd and rpc.mountd daemons. Type and dimension: integer-scalar Default: 180 (lower limit: 90 seconds, upper limit: 300 seconds)
Nservers	Specifies the number of concurrent NFS requests the server can handle. Type and dimension: integer-scalar Default: 4 Example: 24

**Table 4-1** Optional attributes

Optional attribute	Description
Protocol	<p>Selects the transport protocol that the NFS server supports. Allowed values are: tcp, udp or all. If you define Protocol to equal all, the NFS server supports both protocols.</p> <p>Type and dimension: string-scalar</p> <p>Default: all</p> <p>Example: "tcp"</p>
Version	<p>This attribute defines the maximum version of NFS that the agent supports.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 3</p> <p>Example: 3</p>

## Resource type definition

```

type NFS (
    static int RestartLimit = 1
    static str ArgList[] = { Nservers, Protocol, LockFileTimeout,
    Version }
    static str Operations = OnOnly
    int Nservers = 4
    str Protocol = all
    int LockFileTimeout = 180
    int Version = 3
)

```

## Sample configurations

On each node in your cluster, you can find sample NFS, NFSRestart, and Share configurations in `/etc/VRTSvc/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\_3, 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:

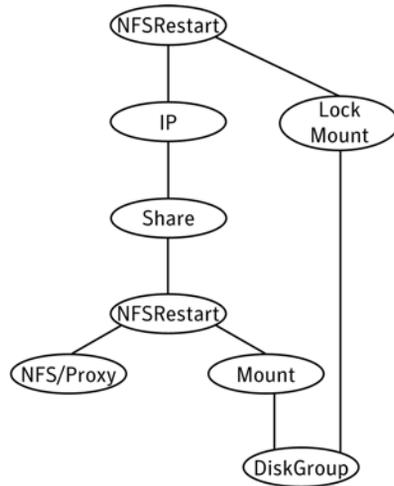
[“NFSRestart agent notes”](#) on page 145

## 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	<p>For the lower NFSRestart resource:</p> <ul style="list-style-type: none"><li>■ Restarts all the NFS daemons that the upper NFSRestart resource stopped previously.</li></ul> <p>For the upper NFSRestart resource:</p> <ul style="list-style-type: none"><li>■ Terminates the statd and lockd daemons to clear the lock state.</li><li>■ Terminates the nfsd and mountd daemons to close the TCP/IP connections.</li><li>■ Terminates the smsyncd daemon if the daemon is running.</li></ul>
Clean	<p>For the lower NFSRestart resource:</p> <ul style="list-style-type: none"><li>■ Restarts all the NFS daemons that the upper NFSRestart resource stopped previously.</li></ul> <p>For the upper NFSRestart resource:</p> <ul style="list-style-type: none"><li>■ Terminates the statd and lockd daemons to clear the lock state.</li><li>■ Terminates the nfsd and mountd daemons to close the TCP/IP connections.</li><li>■ Terminates the smsyncd daemon if the daemon is running.</li></ul>
Action	<ul style="list-style-type: none"><li>■ nfsconf.vfd Checks the runlevel information of the system service nfslock to confirm that the lock daemons do not come online automatically after reboot.</li><li>■ lockdir.vfd Verifies that the NFS lock directory (which is specified by the LocksPathName attribute of NFSRestart) is on shared storage.</li></ul>

## 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. Type and dimension: string-scalar Example: nfs_res1

**Table 4-3** Optional attributes

Optional attribute	Description
LocksPathName	The path name of the directory to store the NFS lock information for all the shared filesystems. This attribute is mandatory when 'NFSLockFailOver = 1'. 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 keylist SupportedActions = { "lockdir.vfd", "nfsconf.vfd"
  }
  static str ArgList[] = { "NFSRes:LockFileTimeout",
  "NFSRes:Nservers", "NFSRes:Version", LocksPathName,
  NFSLockFailover, Lower, State }
  str NFSRes
  int Lower = 0
  str LocksPathName
  boolean NFSLockFailover = 0
)
```

## NFSRestart agent notes

The NFSRestart agent has the following notes:

- [“About high availability fire drill”](#) on page 145
- [“Providing a fully qualified host name”](#) on page 145

### 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, nfserver.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 (nfserver), 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 nsswitch.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/nsswitch.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 `nsswitch.conf` file might affect other services running on the system.

For example:

```
hosts: files [SUCCESS=return] dns 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:

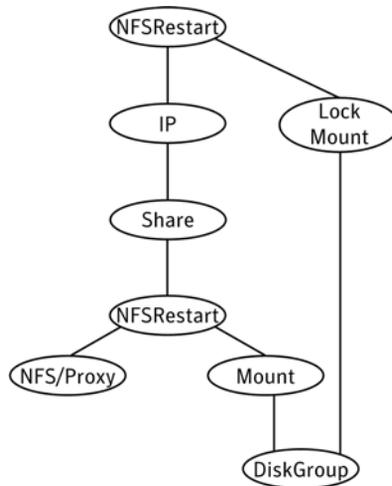
“[Share agent notes](#)” on page 150

## 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	<code>direxists.vfd</code> 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 for the <code>share</code> command. Type and dimension: string-scalar Examples: "-o ro" or "-o rw= <i>hostname</i> "

## Resource type definition

```
type Share (
  static str ArgList[] = { PathName, Options }
  static int NumThreads = 1
  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, DBG\_4, DBG\_5

## 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:  
`/opt/samba/bin`  
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:

```
# smbdc -V
```

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:

```
# smbdc -b | grep CONFIGFILE
```

#### To check for the default location of the Samba pidfile

- ◆ Enter the following command:

```
# smbdc -b | grep PIDDIR
```

## SambaServer agent

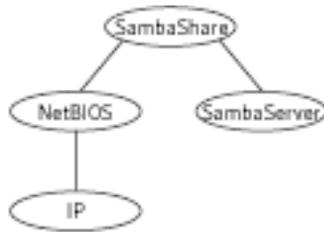
The SambaServer agent starts, stops, and monitors the `smbd` process as a daemon. You can use the agent to make a `smbd` daemon highly available.

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 <code>smbd</code> daemon at specified or default ports.
Offline	Stops the <code>smbd</code> daemon.
Monitor	Verifies that the <code>smbd</code> daemon is running by reading its <code>pid</code> 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 <code>smbd</code> 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/opt/samba/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: "/var/opt/samba/locks"

**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
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

Table 4-7 Optional attributes

Optional attribute	Description
ResponseTimeout	<p>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.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 10</p>
SambaTopDir	<p>Parent path of Samba daemon and binaries.</p> <p>Type and dimension: string-scalar</p> <p>Example: "/opt/samba"</p>
PidFile	<p>The absolute path to the Samba daemon pid file. This file contains the process ID of the monitored smbd process.</p> <p>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 <code>smbd-ConfFile.pid</code> file for monitoring the resource.</p> <p>Type and dimension: string-scalar</p> <p>Example:</p> <p><code>"/var/opt/samba/locks/smbd.pid"</code></p>
SocketAddress	<p>The IP address where the Samba daemon (smbd) listens for connections.</p> <p>Configure the SocketAddress attribute if you are configuring multiple SambaServer resources on a node.</p> <p><b>Note:</b> Only IPv4 addresses are supported.</p> <p>Type and Dimension: string-scalar</p> <p>Example: "10.128.10.14"</p>

## Resource type definitions

```
type SambaServer (  
    static str ArgList[] = { ConfFile, LockDir, Ports,  
        IndepthMonitorCyclePeriod, ResponseTimeout, SambaTopDir,  
        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_SambaServer (  
    ConfFile = "/etc/opt/samba/smb.conf"  
    LockDir = "/var/opt/samba/locks"  
)
```

## Debug log levels

The SambaServer agent uses the following debug log levels:  
DBG\_1, DBG\_2, DBG\_3, DBG\_4, DBG\_5

## SambaShare agent

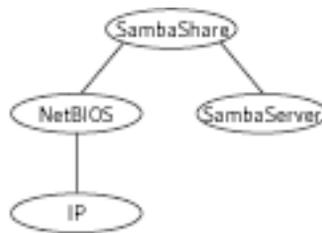
The SambaShare agent adds, removes, and monitors a share by modifying the specified Samba configuration file. You can use the agent to make a Samba Share highly available.

Each filesystem 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 <code>smbclient</code> 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:LockDir", ShareName, ShareOptions,
    "SambaServerRes:Ports", SambaServerRes,
    "SambaServerRes:SambaTopDir", "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.

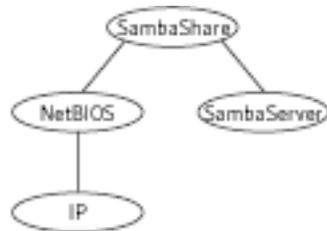
The agent sets, monitors, and resets the names and network interfaces by which the Samba server is known. The agent also sets, monitors and resets Samba to act as a WINS server or domain master or both.

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: <ul style="list-style-type: none"><li>■ NetBIOS name is not advertised.</li><li>■ A NetBIOS alias is not advertised.</li><li>■ Samba is not handling requests on one of the specified interfaces.</li><li>■ If WINS support is configured, Samba is not providing WINS service.</li><li>■ If domain support is set, Samba is not providing Domain Master service.</li></ul>
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	<p>List of network interfaces on which Samba handles browsing.</p> <p>Type and dimension: string-vector</p> <p>Example: "172.29.9.24/16"</p> <p><b>Note:</b> 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.</p>
NetBiosAliases	<p>List of additional names by which the Samba server is known in the network.</p> <p>Type and dimension: string-vector</p> <p>Example: "host1_samba, myname"</p>
WinsSupport	<p>If set to 1, this flag causes the agent to configure Samba as a WINS server.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 0</p>
DomainMaster	<p>If the value of this attribute is 1, the agent sets Samba as Domain Master.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 0</p>

**Table 4-10** Optional attributes

Optional attribute	Description
PidFile	<p>The absolute path to the NetBIOS daemon pid file. This file contains the process ID of the monitored nmbd process.</p> <p>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-<i>ConfFile</i>.pid file for resource monitoring.</p> <p>Type and dimension: string-scalar</p> <p>Example:</p> <p>"/var/opt/samba/locks/nmbd.pid"</p>

## Resource type definition

```

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

# Service and application agents

This chapter contains the following agents:

- [“Apache Web server agent”](#) on page 168
- [“Application agent”](#) on page 178
- [“Process agent”](#) on page 192
- [“ProcessOnOnly agent”](#) on page 198
- [“HPVirtualMachine agent”](#) on page 201
- [“HPVSwitch agent”](#) on page 205

## 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.

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.

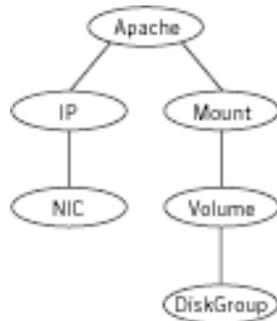
---

For more information regarding this agent:  
See [“Apache Web server notes”](#) on page 174.

## 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	<p>To stop the Apache HTTP server, the agent:</p> <ul style="list-style-type: none"><li>■ Executes the httpdDir/httpd program with the appropriate arguments (Apache v2.0), or</li><li>■ Sends a TERM signal to the HTTP Server parent process (Apache v1.3).</li></ul> <p>When you specify a file with the EnvFile attribute, the file is sourced before the agent executes the httpd command.</p>
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	<p>checkconffile.vfd</p> <p>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.</p>

## State definitions

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

## Attributes

**Table 5-1** Required attributes

Required attribute	Description
ConfigFile	<p>Full path and file name of the main configuration file for the Apache server.</p> <p>Type and dimension: string-scalar</p> <p>Example: "/apache/server1/conf/httpd.conf"</p>
httpdDir	<p>Full path of the directory to the httpd binary file</p> <p>Type and dimension: string-scalar</p> <p>Example: "/apache/server1/bin"</p>
ResLogLevel	<p>Controls the agent's logging detail for a specific instance of a resource. Values are:</p> <ul style="list-style-type: none"> <li>■ ERROR: Logs error messages.</li> <li>■ WARN: Logs error and warning messages.</li> <li>■ INFO: Logs error, warning, and informational messages.</li> <li>■ TRACE: Logs error, warning, informational, and trace messages. Trace logging is verbose. Use for initial configuration or troubleshooting.</li> </ul> <p>Type and dimension: string-scalar</p> <p>Default: INFO</p> <p>Example: "TRACE"</p>
PidFile	<p>This attribute is required when you want to enable the detection of a graceful shutdown outside of VCS control.</p> <p>See "<a href="#">PidFile</a>" on page 172.</p>

Table 5-2 Optional attributes

Optional attribute	Description
DirectiveAfter	<p>A list of directives that httpd processes after reading the configuration file.</p> <p>Type and dimension: string-association</p> <p>Example: DirectiveAfter{} = { KeepAlive=On }</p>
DirectiveBefore	<p>A list of directives that httpd processes before it reads the configuration file.</p> <p>Type and dimension: string-association</p> <p>Example: DirectiveBefore{} = { User=nobody, Group=nobody }</p>
User	<p>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.</p> <p>Type and dimension: string-scalar</p> <p>Example: "apache1"</p>
EnableSSL	<p>Set to 1 (true) to have the online agent function add support for SSL by including the option <code>-DSSL</code> in the start command. For example: <code>/usr/sbin/httpd -f path_to_httpd.conf -k start -DSSL</code></p> <p>Where <code>path_to_httpd.conf</code> file is the path to the <code>httpd.conf</code> file.</p> <p>Set to 0 (false) it excludes the <code>-DSSL</code> option from the command.</p> <p>Type and dimension: boolean-scalar</p> <p>Default: 0</p> <p>Example: "1"</p>

**Table 5-2** Optional attributes

Optional attribute	Description
HostName	<p>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.</p> <p><b>Note:</b> The HostName attribute is only required when the value of SecondLevelMonitor is 1 (true).</p> <p>Type and dimension: string-scalar</p> <p>Example: "web1.example.com"</p>
Port	<p>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).</p> <p>Type and dimension: integer-scalar</p> <p>Default: 80</p> <p>Example: "80"</p>
EnvFile	<p>Full path and file name of the file that is sourced before executing httpdDir/httpd. With Apache 2.0, the file <i>ServerRoot/bin/envvars</i>, 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.</p> <p>Type and dimension: string-scalar</p> <p>Example: "/apache/server1/bin/envvars"</p>
PidFile	<p>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.</p> <p>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).</p> <p>Type and dimension: string-scalar</p> <p>Example: /var/run/httpd.pid</p>

**Table 5-2** Optional attributes

Optional attribute	Description
SharedObjDir	<p>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 <code>-R</code> option when executing the <code>httpd</code> program. Refer to the <code>httpd</code> man pages for more information about the <code>-R</code> option.</p> <p>Type and dimension: boolean-scalar Example: <code>"/apache/server1/libexec"</code></p>
SecondLevelMonitor	<p>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.</p> <p>Type and dimension: boolean-scalar Default: 0 Example: <code>"1"</code></p>
SecondLevelTimeout	<p>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 <code>SecondLevelTimeout</code> 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 <code>MonitorTimeout</code>.</p> <p>Type and dimension: integer-scalar Default: 30</p>

**Table 5-3** Resource type attribute

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

## 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 175
- [“About detecting application failure”](#) on page 175
- [“About bringing an Apache Web server online outside of VCS control”](#) on page 176
- [“About high Availability fire drill”](#) on page 176

## 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 176.
- 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

```
group ApacheG1(  
  SystemList = { host1 = 0, host2 = 1 }  
)  
  
Apache httpd_server (  
  Critical = 0  
  httpdDir = "/apache/bin"  
  HostName = vcshp1  
  Port = 8888  
  User = root  
  SecondLevelMonitor = 1  
  ConfigFile = "/apache/conf/httpd.conf"  
)  
  
DiskGroup Apache_dg (  
  Critical = 0  
  DiskGroup = apc1  
)  
  
IP Apache_ip (  
  Critical = 0  
  Device = lan0  
  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
```

## 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 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.
- 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.
- 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.
- 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`.

## 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	<p>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.</p> <p><b>Note:</b> Do not use the opening and closing ({} ) brace symbols in this string.</p> <p>Type and dimension: string-scalar</p> <p>Example: "/usr/sbin/sample_app start"</p>
StopProgram	<p>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.</p> <p><b>Note:</b> Do not use the opening and closing ({} ) brace symbols in this string.</p> <p>Type and dimension: string-scalar</p> <p>Example: "/usr/sbin/sample_app stop"</p>
At least one of the following attributes: <ul style="list-style-type: none"> <li>■ MonitorProcesses</li> <li>■ MonitorProgram</li> <li>■ PidFiles</li> </ul>	See " <a href="#">Optional attributes</a> " on page 183.

**Table 5-5** Optional attributes

Optional attribute	Description
CleanProgram	<p>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.</p> <p>Type and dimension: string-scalar</p> <p>Example: "/usr/sbin/sample_app force stop"</p>
MonitorProcesses	<p>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.</p> <p>The process name must be the full command line argument that the <code>ps -u user -o args   more</code> command displays for the process.</p> <p>Type and dimension: string-vector</p> <p>Example: "sample_app_process"</p>
MonitorProgram	<p>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.</p> <p>MonitorProgram can return the following VCSAgResState values: 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.</p> <p><b>Note:</b> Do not use the opening and closing ({} ) brace symbols in this string.</p> <p>Type and dimension: string-scalar</p> <p>Example: "/usr/sbin/sample_app_monitor all"</p>

**Table 5-5** Optional attributes

Optional attribute	Description
PidFiles	<p>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.</p> <p>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.</p> <p>Type and dimension: string-vector</p> <p>Example: "/etc/sample/sample_app.pid"</p>
User	<p>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.</p> <p>Type and dimension: string-scalar</p> <p>Default: root</p> <p>Example: user1</p>
EnvFile	<p>The environment file that should get sourced before running any of the StartProgram, StopProgram, MonitorProgram or CleanProgram.</p> <p>Type and dimension: string-scalar</p> <p>Default: ""</p> <p>Example: /home/username/envfile</p>

Table 5-5 Optional attributes

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

## 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 keylist IMFRegList = { MonitorProcesses, User, PidFiles,
    MonitorProgram }
    static keylist RegList = { MonitorProcesses, User }
    str User = root
    str StartProgram
    str StopProgram
    str CleanProgram
    str MonitorProgram
    str PidFiles[]
    str MonitorProcesses[]
    str EnvFile
    boolean UseSUDash
)

```

## 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

### Sample Configuration 1

In this example, configure the executable samba in the StartProgram and StopProgram attributes, with start and stop specified as command-line arguments respectively. Configure the agent to monitor two processes: a process that the smbd.pid specifies, and the process nmbd.

```
Application sendmail (
    User = root
    StartProgram = "/sbin/init.d/sendmail start"
    StopProgram = "/sbin/init.d/sendmail stop"
    PidFiles = {"/etc/mail/sendmail.pid"}
```

```
)
```

## Sample Configuration 2

In this example, since no user is specified, it uses the root user. The executable `samba` starts and stops the application using `start` and `stop` as the command-line arguments. The executable `sambaMonitor` monitors the application and uses `all` as its command-line argument. Also, the agent monitors the `smbd` and `nmbd` processes.

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

## 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 190

## 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 190.

## 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	<p>The FaultTolerance attribute determines when the CoordPoint agent declares that the registrations on the coordination points are missing.</p> <p>If the number of coordination points with missing keys (or registrations) exceeds the value of the FaultTolerance attribute, then the agent reports FAULTED.</p> <p>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).</p> <p>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.</p> <p>To view the current FaultTolerance value, enter the following command:</p> <pre># hares -display coordpoint-res -attribute FaultTolerance</pre> <p>Type and dimension: integer-scalar                      Default: "0"</p>

## 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*.

The agent needs an exact match of PathName and Arguments for all the VCS-managed processes in the process table. To clarify, the processes that the ps command lists should have exact match of PathName and Arguments for all the processes in the VCS configuration file `/etc/VRTSvcs/conf/config/main.cf`.

Note that the AMF kernel driver does not monitor kernel processes. Even if you have enabled intelligent monitoring for Process agent, you must use the traditional poll-based monitoring to monitor kernel processes.

## 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 a process in the background with optional arguments and priority in the specified user context.
Offline	Terminates the process with a <code>SIGTERM</code> . If the process does not terminate, a <code>SIGKILL</code> 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.
<code>imf_init</code>	Initializes the agent to interface with the asynchronous monitoring framework (AMF) kernel driver. This function runs when the agent starts up.
<code>imf_getnotification</code>	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.
<code>imf_register</code>	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	<p>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.</p> <p>This attribute must not exceed 80 characters.</p> <p>Type and dimension: string-scalar</p> <p>Example: "/usr/sbin/sendmail"</p>

**Table 5-9** Optional attributes

Optional attribute	Description
Arguments	<p>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.</p> <p>This attribute must not exceed 80 characters.</p> <p>Type and dimension: string-scalar</p> <p>Example: "bd -q30m"</p>
PidFile	<p>File containing the process ID.</p> <p>Type and dimension: string-scalar</p> <p>Example: "/etc/mail/sendmail.pid"</p>

**Table 5-9** Optional attributes

Optional attribute	Description
Priority	<p>Priority with which the process runs. Effective only when the user is root. Range is 0 to 39 where a process with a priority 0 is the highest.</p> <p>Type and dimension: string-scalar</p> <p>Default: 20</p> <p>Example: "35"</p>
UserName	<p>The user whose ID is used to run the process. The process along with the arguments must run the context of the specified user.</p> <p>Type and dimension: string-scalar</p> <p>Default: root</p> <p>Example: "user1"</p>

## Resource type definition

```

type Process (
    static keylist SupportedActions = { "program.vfd", getcksum }
    static str ArgList[] = { PathName, Arguments, UserName,
        Priority, PidFile }
    str PathName
    str Arguments
    str UserName = root
    str Priority = 20
    str PidFile
)

```

## Sample configurations

### Configuration 1

```

Process sendmail1 (
    PathName = "/usr/sbin/sendmail"
    Arguments = "-bd -q30m"
    User = root
    Priority = 10
    PidFile = "/etc/mail/sendmail.pid"
)

```

## Sample configuration 2

```
cluster vcs_test (
    UserNames = { admin = bIJbIDiFJeJhRJdIG }
    Administrators = { admin }
)

system sysA (
)

system sysB (
)

group ProcessGroup (
    SystemList = { sysa, sysb }
    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	<p>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. Pathname must not exceed 80 characters.</p> <p>The value configured for this attribute needs to match the process listing from the ps output for the agent to display as ONLINE.</p> <p>Type and dimension: string-scalar</p> <p>Example: "/usr/sbin/sendmail"</p>

**Table 5-11** Optional attributes

Optional attribute	Description
IgnoreArgs	<p>A flag that indicates whether monitor ignores the argument list.</p> <ul style="list-style-type: none"> <li>■ If the value is 0, it checks the process pathname and argument list.</li> <li>■ If the value is 1, it only checks for the executable pathname and ignores the rest of the argument list.</li> </ul> <p>Type and dimension: boolean-scalar</p> <p>Default: 0</p>
Arguments	<p>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 a total of 80 characters.</p> <p>Type and dimension: string-scalar</p> <p>Example: "-bd -q30m"</p>

## Resource type definition

```
type ProcessOnOnly (
    static str ArgList[] = { IgnoreArgs, PathName, Arguments }
    static str Operations = OnOnly
    boolean IgnoreArgs = 0
    str PathName
    str Arguments
)
```

## Debug log levels

The ProcessOnOnly agent uses the following debug log levels:  
DBG\_1

# HPVirtualMachine agent

The HPVirtualMachine agent brings online, takes offline, and monitors virtual machines (VMGuests) that are running on the physical host (VMHost).

## Limitations

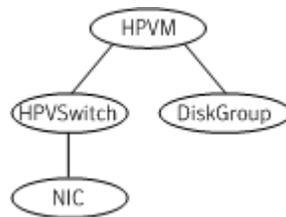
The HPVirtualMachine agent has the following limitations:

- The agent cannot detect if the operating system hangs. Even if the HPVirtualMachine reports the VMGuest state as `ONLINE`, it does not mean that the OS running within guest is functioning properly.

## Dependencies

This resource depends on the DiskGroup agent and the HPVSwitch agent resources for its datastore and network.

**Figure 5-4** Sample service group for a HPVirtualMachine resource, where HPVM represents the HPVirtualMachine resource



## Agent functions

Online	Uses the <code>hvvmstart</code> command to start the virtual machine (VMGuest).
Offline	Attempts a graceful shut down of the virtual machine. Uses the <code>hvvmstop</code> command to stop the VMGuest.
Monitor	Uses the <code>hvvmstatus</code> command to detect the virtual machine's state. Returns the following status: <ul style="list-style-type: none"> <li>■ If the Virtual Machine is missing, it returns an <code>UNKNOWN</code> state.</li> <li>■ If the Virtual Machine is running, then it returns an <code>ONLINE</code> state.</li> <li>■ If the Virtual Machine is not running, it returns <code>OFFLINE</code>.</li> </ul>
Clean	Forcefully shuts down the virtual machine. It uses <code>hvvmstop</code> command with <code>-h</code> argument.

## State definitions

ONLINE	Indicates that the virtual machine (VMGuest) is up and has a heartbeat.
OFFLINE	Indicates that the virtual machine is turned off.
FAULTED	Indicates that the virtual machine has failed to start up using the online operation. This can occur due to an issue with the VMGuest configuration. It can also occur due to a sudden unexpected shutdown of the virtual machine.
UNKNOWN	Indicates the agent cannot determine the virtual machine's state. This state can occur if the virtual machine has not been created yet or the resource type definition of the HPVirtualMachine agent is not configured properly.

## Attributes

**Table 5-12** Required attributes

Required attribute	Description
VMName	The virtual machine (VMGuest) name that the agent monitors. This attribute is unique. Type-dimension: string-scalar Example: "vmg_01"
DelayAfterOnline	Defines the maximum time that the VMGuest can take to reach the EFI (Extensible Firmware Interface) shell. If the VMGuest leaves the EFI shell before the time provided by DelayAfterOnline, the online function exits at that time only. This attribute is added as to ensure that the boot time of VMGuest up to EFI shell remains flexible and can be modified as per the requirement. Type-dimension: integer-scalar Default: 30

**Table 5-13** Resource type attribute

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

## Resource type definition

```

type HPVirtualMachine (
    static int IntentionalOffline = 1
    static str ArgList[] = { VMName, DelayAfterOnline, State, IState
    }
    str VMName
    int DelayAfterOnline = 30
)

```

## Sample configurations

### Basic HPVirtualMachine configuration

```
group group1 (  
  SystemList = { system1 = 0, system = 2 }  
)  
HPVirtualMachine vm (  
  VMName = vcsivm  
)
```

### HPVirtualMachine configuration with an HPVSwitch resource

```
group group1 (  
  SystemList = { system1 = 0, system = 2 }  
)  
HPVirtualMachine vm (  
  VMName = vcsivm  
)  
  
HPVSwitch vswitch (  
  VSwitchName = pub0  
)  
vm requires vswitch
```

## Debug log levels

The HPVirtualMachine agent uses the following debug log levels:  
DBG\_1, DBG\_3, DBG\_4, DBG\_5

## HPVSwitch agent

Use the HPVSwitch agent to manage and control the virtual switches that are associated with the network connection of the virtual machines (VMGuests). Virtual switches are virtual entities that resemble a normal switch to VMGuests. Virtual switches are mapped to the physical network interface card (NIC) on the physical machine (VMHost).

---

**Note:** If the virtual switch is started upon system reboot, you may observe a concurrency violation for the HPVSwitch resource that is configured as a part of a failover service group.

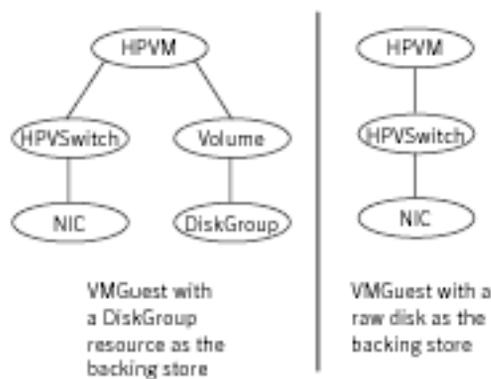
---

For agent limitations and requirements, see:  
“[HPVSwitch agent notes](#)” on page 207

## Dependencies

Virtual switches are mapped to the physical NIC on the physical machine (VMHost). The HPVSwitch resource depends on the NIC resource for its functionality. In the following diagrams, HPVM represents the HPVirtualMachine resource.

**Figure 5-5** Sample service group for HPVSwitch resources



## Agent functions

Online	Starts the virtual switch using the native <code>hvvmnet</code> commands. The online function requires the <code>VSwitchName</code> attribute to operate. When a switch is started, all the virtual NICs that are mapped to that virtual switch are activated.
Offline	Stops the virtual switch using the <code>hvvmnet</code> command. After the shutdown of VMGuest, no network is required. The native <code>hvvmnet</code> command halts the VSwitch.
Monitor	This function checks the status of the VSwitch and returns the following results: <ul style="list-style-type: none"> <li>■ If the virtual switch is running, it returns <code>ONLINE</code>.</li> <li>■ If the virtual switch is not running or is halted, it returns <code>OFFLINE</code>.</li> <li>■ If the virtual switch is not present on the system, it returns the state <code>UNKNOWN</code>.</li> </ul>
Clean	Stops the virtual switch. It is similar to the offline operation.

## State definitions

<code>ONLINE</code>	The VSwitch for the HPVSwitch agent is running properly.
<code>OFFLINE</code>	The VSwitch is halted. It is currently switched off.
<code>UNKNOWN</code>	The VSwitch is not configured properly.
<code>FAULTED</code>	The VSwitch has unexpectedly turned off or it failed to start.

## Attributes

**Table 5-14** Required attributes

Required attribute	Description
<code>VSwitchName</code>	Name of the VSwitch that the HPVSwitch agent monitors. This attribute is unique. Type-dimension: string-scalar Example: "vswitch_01"

## HPVSwitch agent notes

The HPVSwitch agent has the following notes:

- [“Agent limitation”](#) on page 207
- [“Requirements”](#) on page 207

### Agent limitation

This agent does not manage the virtual NIC on the VMGuest.

### Requirements

For the HPVSwitch resource agents to work, the virtual switches must be present on the VMHost because the VMHost controls the virtual switches. The VMGuest has no information about the virtual switches.

## Resource type definition

```
type HPVSwitch (  
    static str ArgList[] = { VSwitchName }  
    str VSwitchName  
)
```

## Sample configurations

The following sample configurations are for HP Integrity virtual machine service groups:

- [“Creating an Integrity virtual machine service group”](#) on page 207
- [“Service group with raw VxVM volume as the backing store”](#) on page 208
- [“Service group with raw disk as the backing store”](#) on page 209
- [“Service group with an online VM guest migration feature”](#) on page 210

### Creating an Integrity virtual machine service group

The following is a high-level overview of creating an Integrity virtual machine service group.

#### Configuring the failover service group

Configure a failover service group. Perform the following steps:

- Name the service group *IVM*.
- Make sure that the SystemList attribute contains all the cluster nodes where the VMGuest can failover.

### Creating an HP virtual switch resource

Create and configure an HPVSwitch resource. Perform the following steps:

- Configure an HPVSwitch resource *vswitch* inside the IVM service group.
- Assign the virtual switch name that you want to monitor in the VSwitchName attribute.
- Create a NIC resource *nic* inside the IVM service group.
- Make sure that the Device attribute contains the virtual switch's backing physical NIC.
- Create a dependency between the parent HPVSwitch resource *vswitch* and its child NIC resource *nic*.

### Creating the VMGuest resource

Create and configure the VMGuest resource. Perform the following steps:

- Configure a HPVVirtualMachine resource *hpvm* inside the IVM service group.
- Add the name of the VMGuest that you want to monitor in the VMName attribute.
- Create a dependency between the parent HPVVirtualMachine resource *hpvm* and its child resource *vswitch*.

### Creating backing storage resources for the VMGuest resource

Create and configure the backing storage resources. Perform the following steps:

- If you intend to use a disk group as the backing store for the VMGuest, configure a DiskGroup resource *dg* with the disk group's name in the DiskGroup attribute. If you are using LVMs, raw disks, or CVM, use the corresponding storage resources.
- If the VMGuest is installed on a VxVM volume, create a Volume resource *vol*. Use the volume name for the backing storage for the VMGuest. Refer to the *hpvm* commands to learn more about backing storage for VMGuests.
- Create a dependency between the parent HPVVirtualMachine resource *vm* and the child Volume resource *vol*.
- Ensure that all resources are enabled before bringing them online.

### Service group with raw VxVM volume as the backing store

```
include "types.cf"
cluster ivmclus (
)
system sysA (
```

```

system sysB (
)
group IVM (
  SystemList = { sysA = 0, sysB = 1 }
)
  DiskGroup dg (
    DiskGroup = dg1
  )
  HPVSwitch pub0 (
    VswitchName = pub0
  )
  HPVirtualMachine hpvm (
    VMName = vcsivml
  )
  NIC nic (
    Device = lan0
  )
  Volume vol (
    Volume = vol1
    DiskGroup = dg1
  )
  pub0 requires nic
  vm requires pub0
  vm requires vol
  vol requires dg

// resource dependency tree
//
//   group IVM_OS
//   {
//     HPVirtualMachine hpvm
//     {
//       Volume vol
//       {
//         DiskGroup dg
//       }
//     }
//     HPVSwitch pub0
//     {
//       NIC nic
//     }
//   }
// }

```

## Service group with raw disk as the backing store

```

include "types.cf"
cluster Test (
)
  system SysA (
  )
  system SysB (
  )

```

```
group g1 (
  SystemList = { SysA = 0, SysB = 1 }
)
HPVSwitch vswitch (
  VSwitchName = public0
)
HPVirtualMachine hpvm (
  VMName = vmsharedhp
)
hpvm requires vswitch
```

## Service group with an online VM guest migration feature

```
include "types.cf"
cluster ivmclus (
)
system sysA (
)
system sysB (
)
group IVM (
  SystemList = { sysA = 0, sysB = 1 }
)
HPVirtualMachine hpvm (
  VMName = vcsivm1
)
```

## Debug log levels

The HPVSwitch agent uses the following debug log levels:

DBG\_1, DBG\_3, DBG\_4

# Infrastructure and support agents

This chapter contains the following agents:

- [“About the infrastructure and support agents”](#) on page 211
- [“NotifierMngr agent”](#) on page 212
- [“Proxy agent”](#) on page 219
- [“Phantom agent”](#) on page 223
- [“RemoteGroup agent”](#) on page 225

## 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 <code>SIGABORT</code> . If the process does not exit within one second, VCS sends a <code>SIGKILL</code> .
Monitor	Monitors the notifier process.
Clean	Sends <code>SIGKILL</code> .

### 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	<p>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.</p> <p><b>Note:</b> SnmpConsoles is a required attribute if SmtServer is not specified; otherwise, SnmpConsoles is an optional attribute. Specify both SnmpConsoles and SmtServer if desired.</p> <p>Type and dimension: string-association</p> <p>Example:                      "172.29.10.89" = Error, "172.29.10.56" = Information</p>
SmtServer	<p>Specifies the machine name of the SMTP server.</p> <p><b>Note:</b> SmtServer is a required attribute if SnmpConsoles is not specified; otherwise, SmtServer is an optional attribute. You can specify both SmtServer and SnmpConsoles.</p> <p>Type and dimension: string-scalar</p> <p>Example: "smtp.example.com"</p>

**Table 6-2** Optional attributes

Optional attribute	Description
EngineListeningPort	<p>Change this attribute if the VCS engine is listening on a port other than its default port.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 14141</p>

**Table 6-2** Optional attributes

Optional attribute	Description
MessagesQueue	<p>Size of the VCS engine's message queue. Minimum value is 30.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 30</p>
NotifierListeningPort	<p>Any valid, unused TCP/IP port number.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 14144</p>
NotifierSourceIP	<p>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.</p> <p><b>Note:</b> Make sure that the SourceIP given in this attribute is present in the /etc/hosts file or is DNS-resolvable.</p> <p>Type and dimension: string-scalar</p> <p>Example: "10.209.77.111"</p>
SmtplibFromPath	<p>Set to a valid email address, if you want the notifier to use a custom email address in the FROM: field.</p> <p>Type and dimension: string-scalar</p> <p>Example: "usera@example.com"</p>

**Table 6-2** Optional attributes

Optional attribute	Description
Smtprcipients	<p>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.</p> <p><b>Note:</b> Smtprcipients is a required attribute if you specify Smtprserver.</p> <p>Type and dimension: string-association</p> <p>Example:                      "james@example.com" = SevereError,                      "admin@example.com" = Warning</p>
Smtprturnpath	<p>Set to a valid email address, if you want the notifier to use a custom email address in the Return-Path: &lt;&gt; field.</p> <p>If the mail server specified in Smtprserver does not support SMTP VRFY command, then set the Smtprvfyoff to 1 in order for the Smtprturnpath value to take effect.</p> <p>Type and dimension: string-scalar</p> <p>Example: "usera@example.com"</p>
Smtprservertimeout	<p>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. If you notice that the mail server is taking a longer duration to reply back to the SMTP commands sent by notifier, you can increase this value.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 10</p>

**Table 6-2** Optional attributes

Optional attribute	Description
SmtptServerVrfyOff	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 the SmtptServer 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
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

## Resource type definition

```

type NotifierMngr (
    static int RestartLimit = 3
    static str ArgList[] = { EngineListeningPort, MessagesQueue,
    NotifierListeningPort, NotifierSourceIP, SnmpdTrapPort,
    SnmpCommunity, SnmpConsoles, SmtptServer, SmtptServerVrfyOff,
    SmtptServerTimeout, SmtptReturnPath, SmtptFromPath, SmtptRecipients
    }
    int EngineListeningPort = 14141
    int MessagesQueue = 30
    int NotifierListeningPort = 14144
    int SnmpdTrapPort = 162
    str SnmpCommunity = public
    str SnmpConsoles{}
    str SmtptServer
    boolean SmtptServerVrfyOff = 0
    int SmtptServerTimeout = 10
    str SmtptReturnPath

```

```
    str SmtFromPath
    str SmtpRecipients{}
    str NotifierSourceIP
)
```

## 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 (`vcadmin@example.com`).

## Configuration

```
system north

system south

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

Phantom my_phantom (
)

NIC NicGrp_en0 (
    Device = lan0
    NetworkHosts = { "166.93.2.1", "166.97.1.2" }
)

group Grp1 (
    SystemList = { north, south }
```

```
AutoStartList = { north }
)

Proxy nicproxy(
TargetResName = "NicGrp_en0"
)

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

ntfr requires nicproxy

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

## Proxy agent

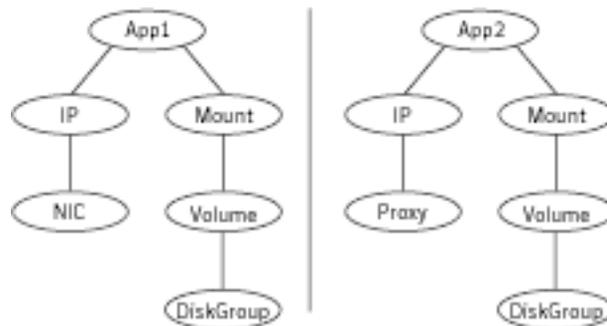
The Proxy agent mirrors the state of another resource on a local or remote system. It provides a means to specify and modify one resource and have its state reflected by its proxies. You can use the agent when you need to replicate the status of a resource.

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	<p>Name of the target resource that the Proxy resource mirrors.</p> <p>The target resource must be in a different resource group than the Proxy resource.</p> <p>Type and dimension: string-scalar</p> <p>Example: "tmp_VRTSvcs_file1"</p>

**Table 6-4** Optional attribute

Optional attribute	Description
TargetSysName	<p>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.</p> <p>Type and dimension: string-scalar</p> <p>Example: "sysa"</p>

## 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 = { lan0 = "192.98.16.103",lan3 =
"192.98.16.103" }
    Device@sysb = { lan0 = "192.98.16.104",lan3 =
"192.98.16.104" }
    NetMask = "255.255.255.0"
    ArpDelay = 5
    Options = "broadcast 192.203.15.255"
    RouteOptions@sysa = "default 192.98.16.103 0"
```

```
        RouteOptions@sysb = "default 192.98.16.104 0"
    )
    IPMultiNIC ip1 (
        Address = "192.98.14.78"
        NetMask = "255.255.255.0"
        MultiNICResName = mnic
        Options = "broadcast 192.203.15.255"
    )

    ip1 requires mnic

group grp2 (
    SystemList = { sysa, sysb }
    AutoStartList = { sysa }
)
    IPMultiNIC ip2 (
        Address = "192.98.14.79"
        NetMask = "255.255.255.0"
    MultiNICResName = 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 “ <a href="#">ControlMode</a> ” on page 228 for more information.
Offline	Takes the remote service group offline. See the “ <a href="#">ControlMode</a> ” on page 228 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 “ <a href="#">VCSSysName</a> ” on page 227.
Clean	If the RemoteGroup resource faults, the Clean function takes the remote service group offline. See the “ <a href="#">ControlMode</a> ” on page 228 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 “ <a href="#">ReturnIntOffline</a> ” on page 231.
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 “ <a href="#">ReturnIntOffline</a> ” on page 231.
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	<p>The IP address or DNS name of a node in the remote cluster. The IP address can be either physical or virtual.</p> <p>When configuring a virtual IP address of a remote cluster, do not configure the IP resource as a part of the remote service group.</p> <p>Type and dimension: string-scalar</p> <p>Examples: "www.example.com" or "11.183.12.214"</p>
Port	<p>This is a required attribute when the remote cluster listens on a port other than the default value of 14141.</p> <p>See <a href="#">"Port"</a> on page 230.</p>
GroupName	<p>The name of the service group on the remote cluster that you want the RemoteGroup agent to monitor or manage.</p> <p>Type and dimension: string-scalar</p> <p>Example: "DBGp"</p>
VCSSysName	<p>You must set this attribute to either the VCS system name or the ANY value.</p> <ul style="list-style-type: none"> <li>■ ANY The RemoteGroup resource goes online if the remote service group is online on any node in the remote cluster.</li> <li>■ <i>VCSSysName</i> 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.</li> </ul> <p>Type and dimension: string-scalar</p> <p>Example: "vcssys1" or "ANY"</p>

**Table 6-5** Required attributes

Required attribute	Description
ControlMode	<p>Select only one of these values to determine the mode of operation of the RemoteGroup resource: MonitorOnly, OnlineOnly, or OnOff.</p> <ul style="list-style-type: none"> <li>■ <b>OnOff</b>                      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.</li> <li>■ <b>MonitorOnly</b>                      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.</li> <li>■ <b>OnlineOnly</b>                      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.</li> </ul> <p>Type and dimension: string-scalar</p>

**Table 6-5** Required attributes

Required attribute	Description
Username	<p>This is the login user name for the remote cluster.</p> <p>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.</p> <p>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.</p> <p>For a secure remote cluster:</p> <ul style="list-style-type: none"> <li>■ 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.</li> <li>■ 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.</li> </ul> <p>Type and dimension: string-scalar</p> <p>Example:</p> <ul style="list-style-type: none"> <li>■ For a cluster without the Symantec Product Authentication Service: "johnsmith"</li> <li>■ For a secure remote cluster: "foobar@example.com"</li> </ul>
Password	<p>This is the password that corresponds to the user that you specify in the Username attribute. You must encrypt the password with the <code>vcscrypt -agent</code> command.</p> <p><b>Note:</b> Do not use the <code>vcscrypt</code> utility when entering passwords from a configuration wizard or the Cluster Manager (Java Console).</p> <p>Type and dimension: string-scalar</p>

**Table 6-6** Optional attributes

Optional attribute	Description
DomainType	<p>For a secure remote cluster only, enter the domain type information for the specified user.</p> <p>For users who have the domain type unixpwd, you do not have to set this attribute.</p> <p>Type: string-scalar</p> <p>Example: "nis", "nisplus"</p>
BrokerIp	<p>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.</p> <p>Type: string-scalar</p> <p>Example: "128.11.295.51"</p>
Port	<p>The port where the remote engine listens for requests.</p> <p>This is an optional attribute, unless the remote cluster listens on a port other than the default value of 14141.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 14141</p>
OfflineWaitTime	<p>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.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 0</p>

**Table 6-6** Optional attributes

Optional attribute	Description
ReturnIntOffline	<p>Select one of the following values for RemoteGroup to return IntentionalOffline:</p> <ul style="list-style-type: none"> <li>■ RemotePartial—Indicates that the RemoteGroup resource returns an IntentionalOffline if the remote service group is in an ONLINE PARTIAL state.</li> <li>■ RemoteOffline—Indicates that the RemoteGroup resource returns an IntentionalOffline if the remote service group is in an OFFLINE state.</li> <li>■ RemoteFaulted—Indicates that the RemoteGroup resource returns an IntentionalOffline if the remote service group is OFFLINE FAULTED.</li> </ul> <p>You can use these values in combinations with each other.</p> <p>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 <i>Veritas Cluster Server Administrator's Guide</i>.</p> <p>Type and dimension: string-vector                      Default: ""</p>
OfflineMonitoringNode	<p>Defines the cluster node that performs the offline monitoring of the remote service group.</p> <p>This is an internal attribute. Do not modify.</p>

**Table 6-7** Type-level attributes

Type level attributes	Description
OnlineRetryLimit OnlineWaitLimit	<p>In case of remote service groups that take a longer time to Online, Symantec recommends that you modify the default OnlineWaitLimit and OnlineRetryLimit attributes.</p> <p>See the <i>Veritas Cluster Server Administrator's Guide</i> for more information about these attributes.</p>

**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 <i>Veritas Cluster Server Administrator's Guide</i> 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 <i>Veritas Cluster Server Administrator's Guide</i> 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

# Testing agents

This chapter contains the following agents:

- [“About the testing agents”](#) on page 233
- [“ElifNone agent”](#) on page 234
- [“FileNone agent”](#) on page 236
- [“FileOnOff agent”](#) on page 238
- [“FileOnOnly agent”](#) on page 240

## 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

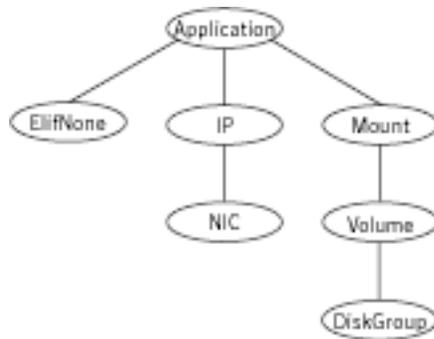
The ElifNone agent monitors a file. It checks for the file's absence.

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

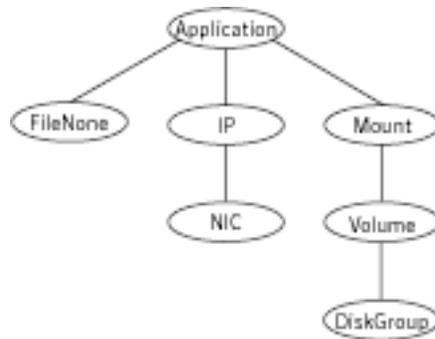
Monitors a file—checks for the file’s existence.

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 str ArgList[] = { PathName }  
    static int OfflineMonitorInterval = 60  
    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

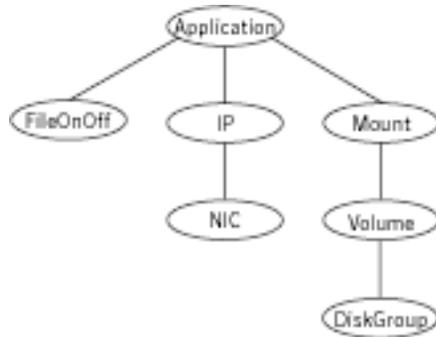
The FileOnOff agent creates, removes, and monitors files.

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

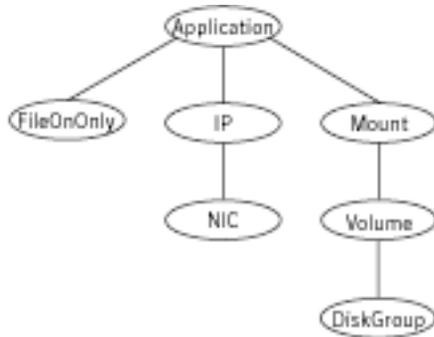
The FileOnOnly agent creates and monitors files.

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

## Numerics

802.1Q trunking 83

## A

about

- Network agents 81
- Samba agents 152

agent

modifying 20

agent functions 149

- Apache Web server agent 169
- Application agent 180
- CoordPoint agent 188
- DiskGroup agent 24
- DiskGroupSnap agent 35
- DNS agent 126
- ElifNone agent 234
- FileNone agent 236
- FileOnOff agent 238
- FileOnOnly agent 240
- HPVirtualMachine agent 201
- HPVSwitch agent 206
- IP agent 85
- IPMultiNIC agent 96
- IPMultiNICB agent 112
- LVMCombo agent 61
- LVMLogicalVolume agent 54
- LVMVolumeGroup agent 58
- Mount agent 66
- MultiNICA agent 102
- MultiNICB agent 118
- NetBIOS agent 162
- NFS agent 138
- NFSRestart agent 142
- NIC agent 91
- NotifierMngr agent 212
- Phantom agent 223
- Process agent 193
- ProcessOnOnly agent 198
- Proxy agent 219
- RemoteGroup agent 226

SambaServer agent 154

SambaShare agent 159

Volume agent 47

VolumeSet agent 51

agents

- Apache Web server 168
- Application 178
- CoordPoint 188
- DiskGroup 24
- DiskGroupSnap 34
- DNS 125
- ElifNone 234
- FileNone 236
- FileOnOff 238
- FileOnOnly 240
- HPVirtualMachine 201
- HPVSwitch 205
- IP 84
- IPMultiNIC 96
- IPMultiNICB 111
- LVMCombo 61
- LVMLogicalVolume 54
- LVMVolumeGroup 58
- Mount 65
- MultiNICA 101
- MultiNICB 116
- NetBIOS 162
- NFS 138
- NFSRestart 141
- NIC 90
- NotifierMngr 212
- Phantom 223
- Process 192
- ProcessOnOnly 198
- Proxy 219
- RemoteGroup 225
- SambaServer 154
- SambaShare 159
- Share 148
- Volume 47
- VolumeSet 51

agents, typical functions 19

- Apache Web server agent
  - agent functions 169
  - attributes 170
  - description 168
  - detecting application failure 175
  - sample configuration 177
  - state definitions 169
- Application agent
  - agent functions 180
  - description 178
  - high availability fire drill 178
  - resource type definition 185
  - sample configurations 186
  - state definitions 181
- association dimension 20
- attribute data types 20
- attributes
  - CoordPoint agent 189
  - DiskGroup agent 27
  - DiskGroupSnap agent 36
  - DNS agent 128
  - ElifNone agent 235
  - FileNone agent 237
  - FileOnOff agent 239
  - FileOnOnly agent 241
  - HPVirtualMachine agent 203
  - HPVSwitch agent 206
  - IP agent 86
  - IPMultiNIC agent 98
  - IPMultiNICB agent 113
  - LVMCombo agent 62
  - LVMLogicalVolume agent 56
  - LVMVolumeGroup agent 59
  - Mount agent 69
  - MultiNICA agent 103
  - MultiNICB agent 119
  - NFS agent 139
  - NFSRestart agent 144
  - NIC agent 92
  - NotifierMngr agent 213
  - Process agent 195
  - Proxy agent 220
  - RemoteGroup agent 227
  - SambaServer agent 156
  - Share agent 150
  - Volume agent 49
  - VolumeSet agent 52
- attributes, modifying 19, 20

**B**

- boolean data types 20
- bundled agents 19

**C**

- Checklist to ensure the proper operation of
  - MultiNICB 110
- Cluster Manager (Java Console), modifying
  - attributes 20
- CNAME record 133
- configuration files
  - main.cf 224
  - modifying 20
  - types.cf 19
- configuring, Samba agents 153
- CoordPoint agent
  - agent functions 188
  - attributes 189
  - description 188
  - resource type definition 190
  - sample configurations 191
  - state definitions 189

**D**

- data type
  - boolean 20
  - string 20
- data types
  - integer 20
- description, resources 19
- dimensions
  - keylist 20
  - scalar 20
  - vector 20
- DiskGroup agent
  - agent functions 24
  - attributes 27
  - description 24
  - high availability fire drill 31
  - resource type definition 31
  - sample configurations 32
  - state definitions 26
- DiskGroupSnap agent
  - agent functions 35
  - attributes 36
  - description 34
  - resource type definition 39
  - sample configurations 40

- state definitions 35
- DNS agent 127
  - agent functions 126
  - attributes 128
  - description 125
  - resource type definition 132
  - sample web server configuration 133

## E

- ElifNone agent
  - agent functions 234
  - attributes 235
  - description 234
  - resource type definition 235
  - sample configuration 235
  - state definitions 234

## F

- FileNone agent
  - agent functions 236
  - attribute 237
  - description 236
  - resource type definition 237
  - sample configurations 237
  - state definitions 236
- FileOnOff agent
  - agent functions 238
  - attribute 239
  - description 238
  - state definitions 239
- FileOnOnly agent
  - agent functions 240
  - attribute 241
  - description 240
  - resource type definition 241
  - sample configuration 241
  - state definitions 240

## H

- high availability fire drill 31, 76, 84, 90, 132, 145, 178, 192
- HP Auto Port Aggregation (APA) support 91, 117
- HPVirtualMachine agent
  - agent functions 201
  - attributes 203
  - description 201
  - resource type definition 203

- sample configurations 204
- state definitions 202
- HPVSwitch agent
  - agent functions 206
  - attributes 206
  - description 205
  - resource type definition 207
  - sample configurations 207
  - state definitions 206

## I

- integer data types 20
- IP agent
  - agent functions 85
  - attributes 86
  - description 84
  - high availability fire drill 84
  - resource type definitions 88
  - sample configurations 88
  - state definitions 85
- IPMultiNIC agent
  - agent functions 96
  - attributes 98
  - description 96
  - resource type definitions 99
  - state definitions 97
- IPMultiNICB agent 115
  - agent functions 112
  - attributes 113
  - description 111
  - manually migrating IP address 115
  - requirements 111
  - resource type definition 115
  - state definitions 112

## K

- keylist dimension 20

## L

- LVMCombo agent
  - agent functions 61
  - attributes 62
  - description 61
  - resource type definition 63
  - sample configurations 63
  - state definitions 62
- LVMLogicalVolume agent

- agent functions 54
- attributes 56
- description 54
- resource type definition 56
- sample configurations 57
- state definitions 55

LVMVolumeGroup agent

- agent functions 58
- attributes 59
- description 58
- resource type definition 59
- sample configurations 59
- state definitions 59

**M**

main.cf 19, 224

main.xml 19

modifying

- configuration files 20

modifying agents 20

monitor scenarios, DNS agent 133

Mount agent

- agent functions 66, 68
- attributes 69
- description 65
- high availability fire drill 76, 132, 145
- notes 76
- resource type definition 75
- sample configurations 79

MultiNICA agent

- agent functions 102
- attributes 103
- description 101
- resource type attributes 106
- RouteOptions, HP-UX 107
- RouteOptions, Solaris 107
- sample configurations 108
- state definitions 102

MultiNICB agent

- agent functions 118
- attributes 119
- description 116
- resource type definition 123
- state definitions 118

**N**

NetBIOS agent

- agent functions 162

- description 162
- resource type definition 163
- sample configurations 165
- state definitions 163

NFS agent

- agent functions 138
- attributes 139
- description 138
- resource type definition 140
- sample configurations 140
- state definitions 139

NFSRestart agent

- agent functions 142
- attributes 144
- description 141
- resource type definition 145
- sample configuration 146
- state definitions 143

NIC agent

- agent functions 91
- attributes 92
- description 90
- high availability fire drill 90
- resource type definitions 93
- sample configurations 94
- state definitions 91

noautoimport flag 32

NotifierMngr agent

- agent functions 212
- attributes 213
- description 212
- resource type definition 216
- sample configurations 217
- state definitions 212

**O**

online query 133

**P**

Phantom agent

- agent functions 223
- description 223
- resource type definition 223
- sample configurations 223

prerequisites

- Samba agents 152

Process agent

- agent functions 193

- attributes 195
- description 192
- high availability fire drill 192
- resource type definition 196
- sample configurations 196
- state definitions 194
- ProcessOnOnly agent
  - agent functions 198
  - description 198
  - resource type definition 200
  - state definitions 198
- Proxy agent
  - agent functions 219
  - attributes 220
  - description 219
  - resource type definition 221
  - sample configurations 221

## R

- RemoteGroup agent
  - agent functions 226
  - attributes 227
  - description 225
  - resource type definition 232
  - state definitions 226
- resource type definition 49
  - SambaShare agent 160
- resource type definitions
  - Application agent 185
  - CoordPoint agent 190
  - DiskGroup agent 31
  - DiskGroupSnap agent 39
  - DNS agent 132
  - ElifNone agent 235
  - FileNone agent 237
  - FileOnOnly agent 241
  - HPVirtualMachine agent 203
  - HPVSwitch agent 207
  - IP agent 88
  - IPMultiNIC agent 99
  - IPMultiNICB agent 115
  - LVMCombo agent 63
  - LVMLogicalVolume agent 56
  - LVMVolumeGroup agent 59
  - Mount agent 75
  - MultiNICA agent 106
  - MultiNICB agent 123
  - NetBIOS agent 163
  - NFS agent 140

- NFSRestart agent 145
- NIC agent 93
- NotifierMngr agent 216
- Phantom agent 223
- Process agent 196
- ProcessOnOnly agent 200
- Proxy agent 221
- RemoteGroup agent 232
- SambaServer agent 158
- Share agent 150
- Volume agent 49
- VolumeSet agent 52
- resource types 19
- resources
  - description of 19

## S

- Samba agents 152
  - overview 152
  - prerequisites 152
- Samba agents configuring 153
- SambaServer agent
  - agent functions 154
  - attributes 156
  - description 154
  - resource type definition 158
  - sample configuration 158
  - state definitions 155
- SambaShare agent 159
  - agent functions 159
  - attributes 160
  - resource type definition 160
  - sample configurations 161
  - state definitions 159
- sample configurations 115
  - Apache Web server agent 177
  - Application agent 186
  - CoordPoint agent 191
  - DiskGroup agent 32
  - DiskGroupSnap agent 40
  - ElifNone agent 235
  - FileNone agent 237
  - FileOnOff agent 239
  - FileOnOnly agent 241
  - HPVirtualMachine agent 204
  - HPVSwitch agent 207
  - IP agent 88
  - IPMultiNICB agent 115
  - LVMCombo agent 63

- LVMLogicalVolume agent 57
- LVMVolumeGroup agent 59
- Mount agent 79
- MultiNICA agent 108
- NetBIOS agent 165
- NFS agent 140
- NFSRestart agent 146
- NIC agent 94
- NotifierMngr agent 217
- Phantom agent 223
- Process agent 196
- Proxy agent 221
- SambaServer agent 158
- SambaShare agent 161
- Share agent 151
- Volume agent 50
- scalar dimension 20
- secure DNS update 133
- Share agent 149
  - agent functions 149
  - attributes 150
  - description 148
  - resource type definitions 150
  - sample configurations 151
  - state definitions 149
- state definitions 52, 127
  - Apache Web server agent 169
  - Application agent 181
  - CoordPoint agent 189
  - DiskGroup agent 26
  - DiskGroupSnap agent 35
  - DNS agent 127
  - ElifNone agent 234
  - FileNone agent 236
  - FileOnOff agent 239
  - FileOnOnly agent 240
  - HPVirtualMachine agent 202
  - HPVSwitch agent 206
  - IP agent 85
  - IPMultiNIC agent 97
  - IPMultiNICB agent 112
  - LVMCombo agent 62
  - LVMLogicalVolume agent 55
  - LVMVolumeGroup agent 59
  - Mount agent 68
  - MultiNICA agent 102
  - MultiNICB agent 118
  - NetBIOS agent 163
  - NFS agent 139

- NFSRestart agent 143
- NIC agent 91
- NotifierMngr agent 212
- Process agent 194
- ProcessOnOnly agent 198
- RemoteGroup agent 226
- SambaServer agent 155
- SambaShare agent 159
- Share agent 149
- Volume agent 48
- VolumeSet agent 52
- string data type 20

## T

- trigger script 123
- trunking 83
- types.cf 19

## V

- VCS, resource types 19
- vector dimension 20
- Volume agent
  - agent functions 47
  - attributes 49
  - description 47
  - sample configurations 50
  - state definitions 48
- VolumeSet agent 52
  - agent functions 51
  - attributes 52
  - description 51
  - resource type definition 52