

# Veritas™ Cluster Server Bundled Agents Reference Guide

HP-UX

6.0

# 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: 6.0

Document version: 6.0.3

## Legal Notice

Copyright © 2015 Symantec Corporation. All rights reserved.

Symantec, the Symantec Logo, the Checkmark Logo, Veritas, Veritas Storage Foundation, CommandCentral, NetBackup, Enterprise Vault, and LiveUpdate 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 software upgrades
- Global support purchased on a regional business hours or 24 hours a day, 7 days a week basis
- Premium service offerings that include Account Management Services

For information about Symantec's support offerings, you can visit our website 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 [customercare\\_apac@symantec.com](mailto:customercare_apac@symantec.com)

Europe, Middle-East, and Africa [semea@symantec.com](mailto:semea@symantec.com)

North America and Latin America [supportolutions@symantec.com](mailto:supportsolutions@symantec.com)

## Documentation

Product guides are available on the media 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.

<https://sort.symantec.com/documents>

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:

[doc\\_feedback@symantec.com](mailto:doc_feedback@symantec.com)

For information regarding the latest HOWTO articles, documentation updates, or to ask a question regarding product documentation, visit the Storage and Clustering Documentation forum on Symantec Connect.

<https://www-secure.symantec.com/connect/storage-management/forums/storage-and-clustering-documentation>

## 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

Technical Support .....	4	
Chapter 1	Introducing Bundled agents .....	17
	About Bundled agents .....	17
	Resources and their attributes .....	18
	Modifying agents and their resources .....	18
	Attributes .....	18
	IMF aware agents .....	19
	Enabling debug log messages .....	19
	VCS support for multi-pathing solutions .....	20
Chapter 2	Storage agents .....	22
	About the storage agents .....	22
	DiskGroup agent .....	22
	Support for online migration for DiskGroup agent .....	23
	Dependencies for DiskGroup agent .....	23
	Agent functions for DiskGroup agent .....	23
	State definitions for DiskGroup agent .....	25
	Attributes for DiskGroup agent .....	25
	Resource type definition for DiskGroup agent .....	28
	Notes for DiskGroup agent .....	29
	Sample configurations for DiskGroup agent .....	30
	Debug log levels for DiskGroup agent .....	31
	DiskGroupSnap agent .....	31
	Dependencies for DiskGroupSnap agent .....	32
	Agent functions for DiskGroupSnap agent .....	33
	State definitions for DiskGroupSnap agent .....	33
	Attributes for DiskGroupSnap agent .....	34
	Notes for DiskGroupSnap agent .....	35
	Resource type definition for DiskGroupSnap agent .....	39
	Sample configurations for DiskGroupSnap agent .....	39
	Debug log levels for DiskGroupSnap agent .....	47
	Volume agent .....	47
	Dependencies for Volume agent .....	47
	Agent functions for Volume agent .....	47

State definitions for Volume agent .....	48
Attributes for Volume agent .....	48
Resource type definition for Volume agent .....	49
Sample configuration for Volume agent .....	49
Debug log levels for Volume agent .....	49
VolumeSet agent .....	49
Dependencies for VolumeSet agent .....	49
Agent functions for VolumeSet agent .....	50
State definitions for VolumeSet agent .....	50
Attributes for VolumeSet agent .....	51
Resource type definition for VolumeSet agent .....	51
Sample configurations for VolumeSet agent .....	51
Agent notes for VolumeSet agent .....	51
Inaccessible volumes prevent the VolumeSet agent from coming online .....	52
Debug log levels for VolumeSet agent .....	52
LVMLogicalVolume agent .....	52
Dependencies for LVMLogicalVolume agent .....	52
Agent functions for LVMLogicalVolume agent .....	53
State definitions for LVMLogicalVolume agent .....	53
Attributes for LVMLogicalVolume agent .....	53
Resource type definition for LVMLogicalVolume agent .....	54
Physical volumes associated with volume groups for LVMLogicalVolume agent .....	54
Sample configurations for LVMLogicalVolume agent .....	55
Debug log levels for LVMLogicalVolume agent .....	55
LVMVolumeGroup agent .....	55
Dependencies for LVMVolumeGroup agent .....	55
Agent functions for LVMVolumeGroup agent .....	56
State definitions for LVMVolumeGroup agent .....	56
Attributes for LVMVolumeGroup agent .....	57
Resource type definition for LVMVolumeGroup agent .....	57
LVMVolumeGroup agent notes .....	57
Sample configurations for LVMVolumeGroup agent .....	57
Debug log levels for LVMVolumeGroup agent .....	58
LVMCombo agent .....	58
Dependencies for LVMCombo agent .....	58
Agent functions for LVMCombo agent .....	59
State definitions for LVMCombo agent .....	59
Attributes for LVMCombo agent .....	60
Resource type definition for LVMCombo agent .....	60
Sample configurations for LVMCombo agent .....	61
Debug log levels for LVMCombo agent .....	61



Mount agent .....	62
Dependencies for Mount agent .....	62
Agent functions for Mount agent .....	62
State definitions for Mount agent .....	64
Attributes for Mount agent .....	65
Resource type definition for Mount agent .....	71
Notes for Mount agent .....	72
High availability fire drill .....	72
VxFS file system lock .....	72
IMF usage notes .....	73
Support for loopback file system for HP-UX .....	73
Enabling Level two monitoring for the Mount agent .....	74
Sample configurations for Mount agent .....	75
Debug log levels for Mount agent .....	76

## Chapter 3

Network agents .....	77
About the network agents .....	77
Agent comparisons .....	77
IP agent .....	79
High availability fire drill for IP agent .....	79
Dependencies for IP agent .....	79
Agent functions for IP agent .....	80
State definitions for IP agent .....	80
Attributes for IP agent .....	80
Resource type definition for IP agent .....	82
Sample configurations for IP agent .....	83
Debug log levels for IP agent .....	84
NIC agent .....	84
High availability fire drill for NIC agent .....	84
Dependencies for NIC agent .....	85
Auto Port Aggregation (APA) support for NIC agent .....	85
Agent functions for NIC agent .....	86
State definitions for NIC agent .....	86
Attributes for NIC agent .....	86
Resource type definition for NIC agent .....	88
Sample configurations for NIC agent .....	88
Debug log levels for NIC agent .....	89
IPMultiNIC agent .....	89
Dependencies for IPMultiNIC agent .....	90
Agent functions for IPMultiNIC agent .....	90
State definitions for IPMultiNIC agent .....	90
Attributes for IPMultiNIC agent .....	91

Resource type definition for IPMultiNIC agent .....	92
Debug log levels .....	93
MultiNICA agent .....	93
Dependencies for MultiNICA agent .....	93
Agent function for MultiNICA agent .....	94
State definitions for MultiNICA agent .....	94
Attributes for MultiNICA agent .....	94
Resource type definition for MultiNICA agent .....	97
Notes for MultiNICA agent .....	98
Using RouteOptions for MultiNICA agent .....	98
Sample configurations for MultiNICA agent .....	99
Debug log levels for MultiNICA agent .....	101
About the IPMultiNICB and MultiNICB agents .....	101
Checklist to ensure the proper operation of MultiNICB .....	101
IPMultiNICB agent .....	101
Dependencies for IPMultiNICB agent .....	102
Requirements for IPMultiNICB .....	102
Agent functions for IPMultiNICB agent .....	102
State definitions for IPMultiNICB agent .....	103
Attributes for IPMultiNICB agent .....	103
Resource type definition for IPMultiNICB agent .....	105
Manually migrating a logical IP address for IPMultiNICB agent .....	106
Sample configurations for IPMultiNICB agent .....	106
Debug log levels for IPMultiNICB agent .....	106
MultiNICB agent .....	106
About the MultiNICB agent .....	107
Failover and failback for MultiNICB agent .....	107
The haping utility for MultiNICB agent .....	108
Auto Port Aggregation (APA) support for MultiNICB agent .....	108
Dependencies for MultiNICB agent .....	108
Agent functions for MultiNICB agent .....	109
State definitions for MultiNICB agent .....	109
Attributes for MultiNICB agent .....	109
Resource type definition for MultiNICB agent .....	113
Trigger script for MultiNICB agent .....	114
IPMultiNICB and MultiNICB configuration for MultiNICB agent .....	114
Debug log levels for MultiNICB agent .....	115
DNS agent .....	115
Dependencies for DNS agent .....	116
Agent functions for DNS agent .....	116
State definitions for DNS agent .....	117

	Attributes for DNS agent .....	118
	Resource type definition for DNS agent .....	124
	Agent notes for DNS agent .....	124
	Sample configurations for DNS agent .....	129
	Debug log levels for DNS agent .....	131
<b>Chapter 4</b>	<b>File share agents .....</b>	<b>132</b>
	About the file service agents .....	132
	NFS agent .....	132
	Dependencies for NFS agent .....	133
	Agent functions for NFS agent .....	133
	State definitions for NFS agent .....	133
	Attributes for NFS agent .....	134
	Resource type definition for NFS agent .....	134
	Sample configurations for NFS agent .....	135
	Debug log levels for NFS agent .....	135
	NFSRestart agent .....	135
	Dependencies for NFSRestart agent .....	135
	Agent functions for NFSRestart agent .....	136
	State definitions for NFSRestart agent .....	137
	Attributes for NFSRestart agent .....	138
	Resource type definition for NFSRestart agent .....	138
	Notes for NFSRestart agent .....	139
	Sample configurations for NFSRestart agent .....	140
	Debug log levels for NFSRestart agent .....	141
	Share agent .....	141
	Dependencies for Share agent .....	141
	Agent functions for Share agent .....	142
	State definitions for Share agent .....	142
	Attributes for Share agent .....	143
	Resource type definition for Share agent .....	143
	Notes for Share agent .....	143
	Sample configurations for Share agent .....	144
	Debug log levels for Share agent .....	144
	About the Samba agents .....	144
	The Samba agents .....	144
	Before using the Samba agents .....	144
	Supported versions for the Samba agents .....	145
	Notes for configuring the Samba agents .....	145
	SambaServer agent .....	146
	Dependencies for SambaServer agent .....	146
	Agent functions for SambaServer agent .....	146

	State definitions for SambaServer agent .....	146
	Attributes for SambaServer agent .....	147
	Resource type definitions for SambaServer agent .....	148
	Sample configurations for SambaServer agent .....	149
	Debug log levels for SambaServer agent .....	149
	SambaShare agent .....	149
	Dependencies for SambaShare agent .....	149
	Agent functions for SambaShare agent .....	150
	State definitions for SambaShare agent .....	150
	Attributes for SambaShare agent .....	150
	Resource type definition for SambaShare agent .....	151
	Sample configuration for SambaShare agent .....	151
	Debug log levels for SambaShare agent .....	151
	NetBios agent .....	152
	Dependencies for NetBios agent .....	152
	Agent functions for NetBios agent .....	152
	State definitions for NetBios agent .....	153
	Attributes for NetBios agent .....	153
	Resource type definition for NetBios agent .....	155
	Sample configuration for NetBios agent .....	155
	Debug log levels for NetBios agent .....	155
Chapter 5	Service and application agents .....	156
	About the services and applications agents .....	156
	Apache HTTP server agent .....	156
	Dependencies .....	157
	Agent functions .....	157
	State definitions .....	158
	Attributes .....	159
	Resource type definition .....	162
	Apache HTTP server notes .....	163
	Sample configurations .....	165
	Application agent .....	166
	High availability fire drill for Application agent .....	167
	Dependencies for Application agent .....	167
	Agent functions for Application agent .....	168
	State definitions for Application agent .....	170
	Attributes for Application agent .....	171
	Resource type definition for Application agent .....	174
	Notes for Application agent .....	175
	Sample configurations for Application agent .....	176
	Debug log levels for Application agent .....	177

CoordPoint agent .....	177
Dependencies for CoordPoint agent .....	177
Agent functions for CoordPoint agent .....	178
State definitions for CoordPoint agent .....	178
Attributes for CoordPoint agent .....	179
Resource type definition for CoordPoint agent .....	179
Notes for the CoordPoint agent .....	179
Sample configuration for CoordPoint agent .....	180
Debug log levels for CoordPoint agent .....	181
Process agent .....	181
High availability fire drill for Process agent .....	181
Dependencies for Process agent .....	182
Agent functions for Process agent .....	182
State definitions for Process agent .....	183
Attributes for Process agent .....	183
Resource type definition for Process agent .....	184
Usage notes for Process agent .....	185
Sample configurations for Process agent .....	185
Debug log levels for Process agent .....	186
ProcessOnOnly agent .....	186
Dependencies .....	186
Agent functions .....	186
State definitions .....	187
Attributes .....	187
Resource type definition .....	188
ProcessOnOnly agent usage notes .....	188
Debug log levels .....	189
HPVirtualMachine agent .....	189
Limitations .....	189
Dependencies .....	189
Agent functions .....	189
State definitions .....	190
Attributes .....	190
Resource type definition .....	191
Sample configurations .....	191
Debug log levels .....	193
HPVSwitch agent .....	193
Dependencies .....	194
Agent functions .....	194
State definitions .....	195
Attributes .....	195
HPVSwitch agent notes .....	195
Resource type definition .....	195

	Sample configurations .....	196
	Debug log levels .....	199
Chapter 6	Infrastructure and support agents .....	200
	About the infrastructure and support agents .....	200
	NotifierMngr agent .....	200
	Dependency .....	201
	Agent functions .....	201
	State definitions .....	201
	Attributes .....	201
	Resource type definition .....	204
	Sample configuration .....	204
	Proxy agent .....	206
	Dependencies .....	206
	Agent functions .....	207
	Attributes .....	207
	Resource type definition .....	208
	Sample configurations .....	208
	Debug log levels .....	209
	Phantom agent .....	209
	Dependencies .....	210
	Agent functions .....	210
	Attribute .....	210
	Resource type definition .....	210
	Sample configurations .....	211
	RemoteGroup agent .....	211
	Dependency .....	212
	Agent functions .....	212
	State definitions .....	213
	Attributes .....	213
	Resource type definition .....	219
	Debug log levels .....	219
Chapter 7	Testing agents .....	220
	About the testing agents .....	220
	ElifNone agent .....	220
	Dependencies for ElifNone agent .....	220
	Agent function for ElifNone agent .....	221
	State definitions for ElifNone agent .....	221
	Attributes for ElifNone agent .....	221
	Resource type definition for ElifNone agent .....	222
	Sample configuration for ElifNone agent .....	222

Debug log levels for ElifNone agent .....	222
FileNone agent .....	222
Dependencies for FileNone agent .....	222
Agent functions for FileNone agent .....	223
State definitions for FileNone agent .....	223
Attributes for FileNone agent .....	224
Resource type definition for FileNone agent .....	224
Sample configuration for FileNone agent .....	224
Debug log levels for FileNone agent .....	224
FileOnOff agent .....	224
Dependencies for FileOnOff agent .....	225
Agent functions for FileOnOff agent .....	225
State definitions for FileOnOff agent .....	225
Attribute for FileOnOff agent .....	226
Resource type definition for FileOnOff agent .....	226
Sample configuration for FileOnOff agent .....	226
Debug log levels for FileOnOff agent .....	226
FileOnOnly agent .....	226
Dependencies for FileOnOnly agent .....	226
Agent functions for FileOnOnly agent .....	227
State definitions for FileOnOnly agent .....	227
Attribute for FileOnOnly agent .....	228
Resource type definition for FileOnOnly agent .....	228
Sample configuration for FileOnOnly agent .....	228
Debug log levels for FileOnOnly agent .....	228

Chapter 8	Replication agents .....	229
	About the replication agents .....	229
	RVG agent .....	229
	Dependencies for RVG agent .....	230
	Agent functions for RVG agent .....	231
	State definitions for RVG agent .....	231
	Attributes for RVG agent .....	231
	Resource type definitions for RVG agent .....	232
	Sample configurations for RVG agent .....	232
	RVGPrimary agent .....	233
	Dependencies for RVGPrimary agent .....	233
	Agent functions for RVGPrimary agent .....	234
	State definitions for RVGPrimary agent .....	235
	Attributes for RVGPrimary agent .....	235
	Resource type definitions for RVGPrimary agent .....	236
	Sample configurations for RVGPrimary agent .....	236

RVGSnapshot .....	237
Dependencies for RVGSnapshot .....	237
Agent functions for RVGSnapshot .....	238
State definitions for RVGSnapshot .....	238
Attributes for RVGSnapshot .....	238
Resource type definitions for RVGSnapshot .....	239
Sample configurations for RVGSnapshot .....	239
RVGShared agent .....	240
Dependencies for RVGShared agent .....	240
Agent functions for RVGShared agent .....	240
State definitions for RVGShared agent .....	241
Attributes for RVGShared agent .....	241
Resource type definitions for RVGShared agent .....	241
Sample configurations for RVGShared agent .....	242
RVGLogowner agent .....	242
Dependencies for RVGLogowner agent .....	242
Agent functions RVGLogowner agent .....	243
State definitions RVGLogowner agent .....	243
Attributes RVGLogowner agent .....	244
Resource type definitions RVGLogowner agent .....	244
RVGLogowner agent notes RVGLogowner agent .....	245
Sample configurations RVGLogowner agent .....	246
RVGSharedPri agent .....	246
Dependencies for RVGSharedPri agent .....	247
Agent functions for RVGSharedPri agent .....	247
State definitions for RVGSharedPri agent .....	248
Attributes for RVGSharedPri agent .....	248
Resource type definitions for RVGSharedPri agent .....	249
Sample configurations for RVGSharedPri agent .....	249
Index .....	250



# Introducing Bundled agents

This chapter includes the following topics:

- [About Bundled agents](#)
- [Resources and their attributes](#)
- [Modifying agents and their resources](#)
- [Attributes](#)
- [IMF aware agents](#)
- [Enabling debug log messages](#)
- [VCS support for multi-pathing solutions](#)

## About Bundled agents

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

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

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

Agents can:

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

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

## Resources and their attributes

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

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

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

## Modifying agents and their resources

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

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

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

## Attributes

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

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

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

**Table 1-1** Attribute data types

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

## IMF aware agents

- Mount agent
- Process agent
- Application agent

## 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 for Mount agent”](#) on page 76.

## VCS support for multi-pathing solutions

This section applies to the following agents:

- LVMVolumeGroup agent
- LVMCombo agent

VCS supports Symantec Dynamic Multi-Pathing (DMP) that is included as a part of the Storage Foundation and High Availability (SFHA) suite of products. Symantec does not support multi-pathing solutions that are not explicitly listed in the hardware compatibility list (HCL). You can find the HCL on the SORT web site, under the Documentation tab. However, Symantec supports third-party solutions, which are included as a part of the operating systems.

Symantec aims to thoroughly test and support third-party and native solutions, but it is not possible to test all third-party multi-pathing applications. This is because of complex support matrix and a number of potential product combinations. Hence, Symantec does not officially support multi-pathing solutions that are not explicitly listed in the HCL. Also, advanced functionality such as I/O fencing with SCSI3-PGR is only supported with arrays and multi-pathing solutions listed in the HCL and only with Symantec Storage Foundation.

If you are using a third-party multi-pathing solution, Symantec understands your need of keeping data paths redundant and does not insist that you uninstall or disable the solution. Symantec does not consider third-party multi-pathing solutions as invalid and continues to troubleshoot any support issues. However, for persisting support issues related to multi-pathing solutions, you need to contact the multi-pathing vendor.

# Storage agents

This chapter includes the following topics:

- [About the storage agents](#)
- [DiskGroup agent](#)
- [DiskGroupSnap agent](#)
- [Volume agent](#)
- [VolumeSet agent](#)
- [LVMLogicalVolume agent](#)
- [LVMVolumeGroup agent](#)
- [LVMCombo agent](#)
- [Mount agent](#)

## About the storage agents

Storage agents monitor shared storage and make shared storage highly available. Storage includes shared disks, disk groups, volumes, and mounts.

## DiskGroup agent

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

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

For important information on this agent, See [“Notes for DiskGroup agent”](#) on page 29.

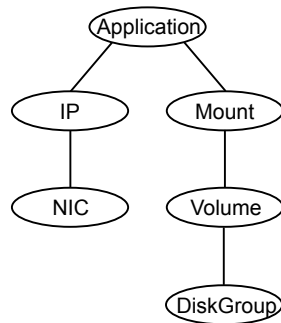
## Support for online migration for DiskGroup agent

VCS supports online migration of data from LVM volumes to VxVM volumes in SFHA environment. For more details, refer to the *SFHA solutions Guide*.

## Dependencies for DiskGroup agent

The DiskGroup resource does not depend on any other resources.

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



## Agent functions for DiskGroup agent

Online	Imports the disk group using the <code>vxldg</code> command.
Offline	Deports the disk group using the <code>vxldg</code> command.
Monitor	Determines if the disk group is online or offline using the <code>vxldg</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.  <pre># vxldg -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</p> <pre>DiskType auto:cdsdisk FreeSize 12765712</pre>
	<p>The value specified is in kilo bytes.</p>
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 diskgroup are visible on host—if it fails, check if the path to disks exists from the host and check if LUN masking and zoning are set properly.</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 29.</p>



## State definitions for DiskGroup agent

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 for DiskGroup agent

**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 service group containing the resource offline. Type and dimension: boolean-scalar Default: 0 <b>Note:</b> If the MonitorReservation attribute is set to 0, the value of the clusterwide attribute UseFence is set to SCSI3, and the disk group is imported without SCSI reservation, then the monitor agent function takes the service group containing the resource offline.

Table 2-2 Optional attributes (*continued*)

Optional attributes	Description
PanicSystemOnDGLoss	<p>Determines whether to panic the node if the disk group becomes disabled or monitor program times out. A loss of storage connectivity can cause the disk group to become disabled. VxVM commands not responding properly can cause monitor program to timeout.</p> <p><b>Note:</b> System administrators may want to set a high value for FaultOnMonitorTimeout to increase system tolerance.</p> <p>This attribute accepts following values 0, 1, 2,3</p> <ul style="list-style-type: none"> <li>■ 0 : Do not halt the system</li> <li>■ 1 : halt the system if either disk group goes into disabled state or the disk group resource faults due to monitor timeout</li> <li>■ 2: halt the system if disk group goes into disabled state</li> <li>■ 3: halt the system if disk group resource faults due to monitor timeout</li> </ul> <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 FAULTED. This state results in the agent attempting to take the service group offline. As part of bringing the DiskGroup resource offline, the agent attempts to deport the disabled disk group. Even if disabled disk group fails to deport, the DiskGroup resource enters a FAULTED state. This state enables the failover of the service group that contains the resource. To fail back the DiskGroup resource, manually deport the disk group after restoring storage connectivity.</li> <li>■ If the cluster does not use I/O fencing, a message is logged and the resource is reported ONLINE. The resource is reported ONLINE so that it does not fail over, which ensures data integrity.</li> </ul> <p><b>Note:</b> The PanicSystemOnDGLoss attribute does not depend on the MonitorReservation attribute.</p> <p><b>Note:</b> If PanicSystemOnDGLoss is set to non-zero value, the system panic is initiated using <code>reboot -hng</code> command. This command halts the system. An administrator needs to bring up the system.</p> <p>Type and dimension: integer-scalar Default: 0</p>

Table 2-2 Optional attributes (continued)

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 of this attribute 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 (*continued*)

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</p> <p>Default: ClusterDefault</p> <p>Example: "SCSI3"</p>

**Table 2-3** Internal attributes

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</p> <p>Default: 1</p>

## Resource type definition for DiskGroup agent

The resource definition for this agent on HP-UX follows:

```
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, ConfidenceLevel }
str DiskGroup
boolean StartVolumes = 1
boolean StopVolumes = 1
boolean MonitorReservation = 0
temp str tempUseFence = INVALID
int PanicSystemOnDGLoss = 0
int UmountVolumes = 0
str Reservation = ClusterDefault
)
```

## Notes for DiskGroup agent

- [High availability fire drill](#)
- [Using volume sets](#)
- [Setting the noautoimport flag for a disk group](#)

### 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 for DiskGroup agent

### 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 for DiskGroup agent

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.

---

**Note:** The DiskGroupSnap agent requires the Global Cluster Option (GCO) license enabled on all systems in the cluster.

---

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

You must define the DiskGroupSnap agent in a separate FireDrill service group which is similar to the Application service group. The FireDrill service group might contain resources similar to the Application service group, for example Mount, Application, and so on.

The FireDrill service group must also contain a resource of type DiskGroupSnap such that the Mount resource depends on the DiskGroupSnap resource. The main DiskGroup must contain multiple sites registered in it with the value of the "siteconsistent" attribute set to on.

When the DiskGroupSnap agent goes online, the agent detaches one of the sites from the main DiskGroup and imports the detached site on the fire drill host as an independent DiskGroup with a different name. The volumes on the DiskGroup are also imported and mounted with same names on the fire drill host.

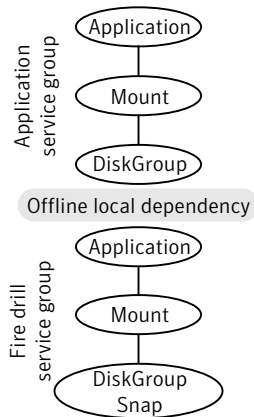
The DiskGroupSnap agent provides Gold and Bronze configurations for the fire drill, which can be specified using the agent's FDType attribute. The configuration decides the site to be detached from the DiskGroup for fire drill testing. The Gold configuration is the default option in which the agent selects a site from the DiskGroup that is neither the local VxVM site nor the site on which the DiskGroup is online. With the Gold configuration, you can also specify an alternate site to detach through the agent's FDSiteName attribute. With the Bronze configuration, the agent uses the local VxVM site name as the site to detach from the DiskGroup.

For important information about this agent, See ["Notes for DiskGroupSnap agent"](#) on page 35.

## Dependencies for DiskGroupSnap agent

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 for DiskGroupSnap agent

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. After the completion of Online function, the agent creates a lock file in the lock directory ( <code>/var/VRTSvcs/lock</code> ) to indicate that the resource is online.
Offline	This re-attaches the site that the value of the FDSiteName attribute specifies back to the application's disk group. After the completion of Offline function the agent removes the lock file from the lock directory ( <code>/var/VRTSvcs/lock</code> ) to indicate that the resource is Offline.
Monitor	Monitors the DiskGroupSnap resource by checking the existence of the Lock file in <code>/var/VRTSvcs/lock</code> directory..
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 for DiskGroupSnap agent

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 for DiskGroupSnap agent

**Table 2-4** Required attributes

Required attribute	Description
TargetResName	The name of the DiskGroup resource from the application service group. Type-dimension: string-scalar Example: "dgres"
FDType	Specifies the configuration to be used for the fire drill. The possible values for this attribute are: <ul style="list-style-type: none"><li>■ Bronze</li><li>■ Gold (default)</li></ul> The Bronze configuration uses the local host's VxVM site name as the site to be detached from the DiskGroup. This action leaves the DiskGroup vulnerable to site disaster since a copy of the production volume might not be available when the fire drill is in progress.  In the Gold configuration there are at least three copies of the parent volume available on different sites, hence, even after detaching one site the volume is not vulnerable to site disaster while the fire drill is in progress.

Table 2-5 Optional attributes

Optional attribute	Description
FDSiteName	<p>The unique VxVM site name tag for the fire drill disks. The value of this attribute is used in conjunction with the FDType attribute and it must be set to one of the sites registered in the main DiskGroup.</p> <ul style="list-style-type: none"> <li>When FDType is set to the Bronze configuration, the value of FDSiteName should either be empty or the name of the local host VxVM site for the fire drill host.</li> <li>When FDType is set to the Gold configuration, FDSiteName identifies a site in the DiskGroup to detach as a part of the fire drill. If FDSiteName is left blank, the agent will choose a site to detach based on the DiskGroup configuration. The agent chooses a site name from the DiskGroup which is neither the production server's site name nor the fire drill host's site name.</li> </ul> <p><a href="#">Table 2-6</a> shows the possible values of the attributes FDType and FDSiteName and the decision taken by the agent.</p>

Consider a configuration where the Production DiskGroup contains three sites: A, B, and C, and the Application service group is online on a node with local VxVM site ID is A. Fire drill is being done on another node Application service group is online on a node where local VxVM site ID is B.

Table 2-6 Example FDType configurations

FDType	Bronze			Gold/Empty		
FDSitename	Empty	B	C	Empty	B	C
Result	Use B as the site to detach and proceed	Detach site B from DiskGroup	Error	Check if there is another site other than A and B and select it. Else, it is an error	Error	Detach site C from the DiskGroup

## Notes for DiskGroupSnap agent

The DiskGroupSnap agent has the following notes:

- See [“Fire drill configuration after upgrading VCS”](#) on page 36.

- See “[Configuring the SystemZones attribute for the fire drill service group](#)” on page 36.
- See “[Configuring the FireDrill service group](#)” on page 36.
- See “[Adding the ReuseMntPt attribute to the ArgList attribute for the Mount agent type](#)” on page 37.
- See “[Configuration considerations](#)” on page 37.
- See “[Agent limitations](#)” on page 39.

## Fire drill configuration after upgrading VCS

After upgrading VCS from any earlier version to 6.0, delete all resources of type DiskGroupSnap and recreate them again using the new definitions of the attributes. Failure to perform this step might result in an unexpected behavior of the agent.

## 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 fire drill 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 Netlsnr to 1:

```
# haconf -makerw
# hatype -modify Oracle FireDrill 1
# hatype -modify Netlsnr 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:

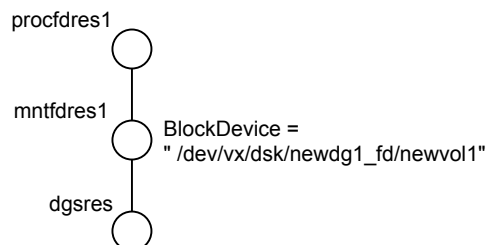
- You must install Veritas Volume Manager 5.1 or later with the FMR license and the Site Awareness license.
- Do not bring the DiskGroupSnap resource online in the SystemZone where the application service group is online.

- 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`. See [Figure 2-3](#) on page 38. shows the changes to resource values for the firedrill service group; note that the `Volume` resource is not included.
- Before commencing the fire drill, make sure that all the sites registered in the application `DiskGroup` are in `ACTIVE` state.

**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 fire drill service group offline before you shut down VCS on any node. If you fail to take the fire drill 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.
- Do not switch the Application service group when fire drill is in progress.

## Resource type definition for DiskGroupSnap agent

The resource type definition for this agent follows:

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

## Sample configurations for DiskGroupSnap agent

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:

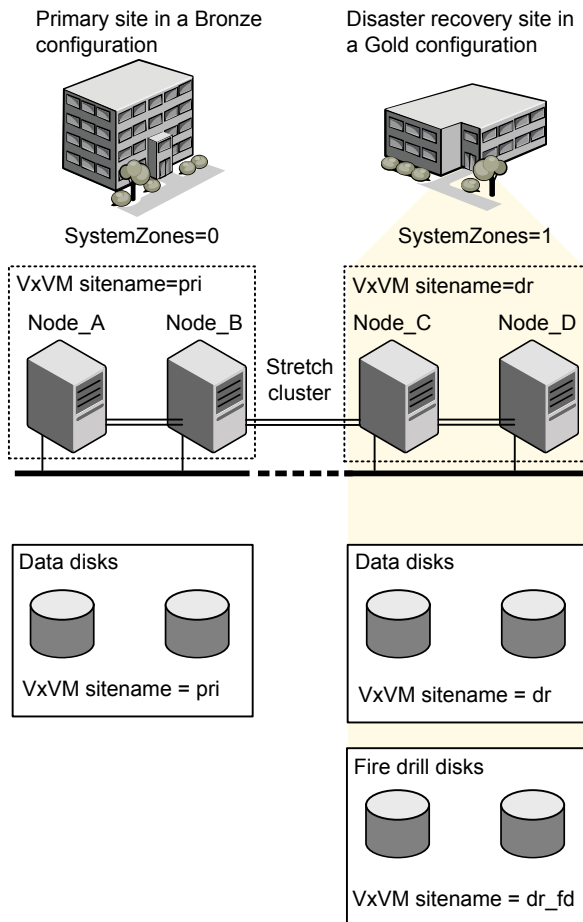
```
FDSiteName@Node_C = dr_fd
FDSiteName@Node_D = dr_fd
```

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 for DiskGroupSnap agent

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
)

FDtype = "Gold"
  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/newdgl_fd/newvol1"
  FSType = vxfs
  FsckOpt = "-y"
  ReuseMntPt = 1
)

Mount mntfdres2 (
  MountPoint = "/dgsfs2"
  BlockDevice = "/dev/vx/dsk/newdgl_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
```

```
procfdres1 requires mntfdres1
procfdres2 requires mntfdres2
```

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

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

DiskGroup dgres (
    DiskGroup = newdgl
)

Mount mntres1 (
    MountPoint = "/dgsfs1"
    BlockDevice = "/dev/vx/dsk/newdgl/newvol1"
    FSType = vxfs
    FsckOpt = "-y"
    ReuseMntPt = 1
)

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

Process procrs1 (
    PathName = "/usr/bin/ksh"
    Arguments = "/scrib.sh /dgsfs1"
)

Process procrs2 (
    PathName = "/usr/bin/ksh"
    Arguments = "/scrib.sh /dgsfs2"
)

mntres1 requires dgres
mntres2 requires dgres
```

```
procrs1 requires mntres1
procrs2 requires mntres2
```

## Sample main.cf of DiskGroupSnap with Oracle resource

The following Oracle configuration has been simplified for presentation within this guide.

```
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
    FDType = "Bronze"
)

IP fd_oraip (
    Device = lan0
    Address = "10.198.95.191"
    NetMask = "255.255.255.0"
)

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

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

NIC fd_oranic (
```

```
Device = lan0
NetworkHosts = { "10.198.95.1" }
)

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 = lan0
    Address = "10.198.95.192"
    Netmask = "255.255.252.0"
)

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 = lan0
)

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 for DiskGroupSnap agent

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.

---

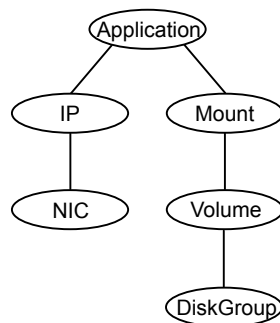
**Note:** Do not use the Volume agent for volumes created for replication.

---

## Dependencies for Volume agent

Volume resources depend on DiskGroup resources.

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



## Agent functions for Volume agent

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 for Volume agent

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 for Volume agent

**Table 2-7** 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 2-8** Internal attribute

Optional 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 for Volume agent

The resource type definition for this agent follows:

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

## Sample configuration for Volume agent

The sample configuration for the Volume agent follows:

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

## Debug log levels for Volume agent

The Volume agent uses the following debug log levels:

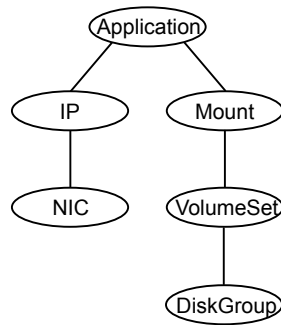
```
DBG_1
```

## 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 for VolumeSet agent

VolumeSet resources depend on DiskGroup resources.

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

## Agent functions for VolumeSet agent

Online	Uses the vxrecover command to start the volume set.
Offline	Uses the vxvset 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 for VolumeSet agent

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 for VolumeSet agent

**Table 2-9** 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 for VolumeSet agent

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

## Sample configurations for VolumeSet agent

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 for VolumeSet agent

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 for VolumeSet agent

The VolumeSet agent uses the following debug log levels:

DBG\_1, DBG\_4

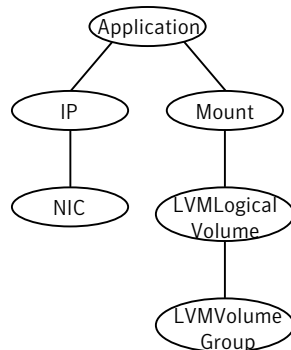
## LVMLogicalVolume agent

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 for LVMLogicalVolume agent

LVMLogicalVolume resources depend on LVMVolumeGroup resources.

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



## Agent functions for LVMLogicalVolume agent

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 for LVMLogicalVolume agent

The state definitions for this agent follow:

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 for LVMLogicalVolume agent

**Table 2-10** 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 2-11** Optional attributes

Optional attribute	Description
VolumeIOTimeout	<p>The time for which the agent should wait before it returns an OFFLINE state when IO 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 for LVMLogicalVolume agent

The resource definition for this agent on HP-UX follows:

```

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 LVMLogicalVolume agent

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 for LVMLogicalVolume agent

The sample configuration for the LVMLogicalVolume agent follows:

### Configuration for LVMLogicalVolume agent

Configuration for HP-UX follows:

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

## Debug log levels for LVMLogicalVolume agent

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.

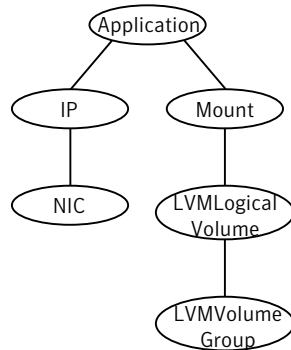
See [“VCS support for multi-pathing solutions”](#) on page 20.

## Dependencies for LVMVolumeGroup agent

No fixed dependencies exist for LVMVolumeGroup Agent. When you create a volume group on disks with single path, Symantec recommends that you use the DiskReservation agent.

With Veritas Dynamic Multi-Pathing, do not use the DiskReservation agent.

The LVMVolumeGroup resource has no dependencies.

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

## Agent functions for LVMVolumeGroup agent

The agent functions for this agent follow:

Online	Activates a volume group. While each system in the cluster must import the volume group, each system does not need to activate it.  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	HP-UX: 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 for LVMVolumeGroup agent

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 for LVMVolumeGroup agent

**Table 2-12** 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 for LVMVolumeGroup agent

The resource definition for this agent on HP-UX follows:

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

## LVMVolumeGroup agent notes

The LVMVolumeGroup agent has the following notes:

## Sample configurations for LVMVolumeGroup agent

The sample configurations for this agent follow:

### Configuration 1

Configuration 1 for HP-UX follows:

```
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 for LVMVolumeGroup agent

The LVMVolumeGroup agent uses the following debug log levels:

DBG\_1

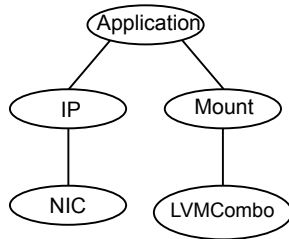
## LVMCombo agent

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.

See [“VCS support for multi-pathing solutions”](#) on page 20.

## Dependencies for LVMCombo agent

No dependencies exist for the LVMCombo resource.

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

## Agent functions for LVMCombo agent

The agent functions for this agent follow:

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 for LVMCombo agent

The state definitions for this agent follow:

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 for LVMCombo agent

**Table 2-13** 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 2-14** Optional attributes

Optional Attribute	Description
VolumeIOTimeout	The time for which the agent waits before it returns an OFFLINE state when I/Os 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 for LVMCombo agent

The resource definition for this agent on HP-UX follows:

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

```
int VolumeIOTimeout = 15
)
```

## Sample configurations for LVMCombo agent

The sample configurations for the LVMCombo agent:

### Sample 1 for LVMCombo agent

```
LVMCombo vg01 (
    VolumeGroup = vg01
    LogicalVolumes = { lv01, lv02 }
)
```

### 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 = { lv01 }
)

Mount mres (
    MountPoint = "/dir2"
    BlockDevice = "/dev/vg02/lv01"
    FSType = vxfs
    MountOpt = ro
    FsckOpt = "-y"
)

mres requires lvmcbres
```

## Debug log levels for LVMCombo agent

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 for mounts is supported only for VxFS and NFS filesystem types.

---

This agent also supports high availability fire drills.

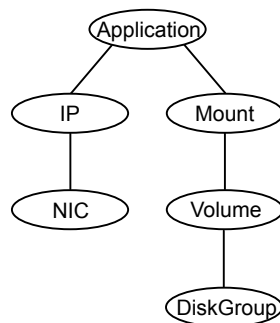
For important information about this agent, See [“Notes for Mount agent”](#) on page 72.

## Dependencies for Mount agent

The Mount resource does not depend on any other resources.

In some scenarios, the Mount agent depends on Volume or DiskGroup resources.

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



## Agent functions for Mount agent

### Online

Mounts a block device on the directory. If the mount process fails for non-NFS mounts, the agent attempts to run the fsck command on the device before attempting to mount the file system again.

If file system type is NFS, agent mounts the remote file system to a specified directory. The remote NFS file system is specified in the BlockDevice attribute.

Offline	Unmounts the mounted file system.
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>
imf_init	Initializes the agent to interface with the asynchronous monitoring framework (AMF) kernel driver. This function runs when the agent starts up.
imf_getnotifi cation	Waits for notification about resource state changes. This function runs after the agent initializes with the AMF kernel driver. The agent continuously waits for notification and takes action on the resource upon notification.
imf_register	Registers the resource entities, which the agent must monitor, with the AMF kernel driver. This function runs for each resource after the resource goes into steady state (online or offline).This action entry point registers mountpoint, blockdevice, and fstype for mount agent.
Clean	Unmounts the mounted file system forcefully.
Info	<p>The Mount agent info 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</pre> <pre># 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>■ <code>chgmtlock</code> Resets the VxFS file system lock to a VCS-defined lock.</li><li>■ <code>mountpoint.vfd</code> 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>■ <code>mounted.vfd</code> 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>■ <code>vxfslic.vfd</code> Checks for valid Veritas File System (VxFS) licenses. If it fails, you need to update the license for VxFS.</li><li>■ <code>mountentry.vfd</code> 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>
<code>attr_changed</code>	Unlocks the mounts when you change the value of the <code>VxFSMountLock</code> attribute from 1 (or 2) to 0 and vice-versa.

## State definitions for Mount agent

The state definitions for this agent follow:

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 file system 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 file system 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 file system has unexpectedly unmounted.</p>
UNKNOWN	Indicates that a problem exists either with the configuration or the ability to determine the status of the resource.



## Attributes for Mount agent

**Table 2-15** Required attributes

Required attribute	Description
BlockDevice	<p>Block device for mount point.</p> <p>When you specify the block device to mount NFS remote file system with IPv6 addresses, enclose IPv6 addresses in square brackets. The mount command requires square brackets around the IPv6 address to differentiate between the colons (:) in the address and the colon that separates the remote host and remote directory.</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>
FsyncOpt	<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>Both the <code>-y</code> and <code>-n</code> arguments enable the VxFS file systems to perform a log replay before a full <code>fsck</code> operation. <code>-n</code> will check the file system and log it. Read <code>man fsck</code> for further information.</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 FsyncOpt %y</code></p>

**Table 2-15** Required attributes (*continued*)

Required attribute	Description
FSType	Type of file system. Supports vxfs, hfs, lofs, or nfs. Type and dimension: string-scalar Example: "nfs"
MountPoint	Directory on which the BlockDevice needs to be mounted. Type and dimension: string-scalar Example: "/campus1"

Table 2-15 Required attributes (*continued*)

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 agent's monitor 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 agent's monitor 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 2-16 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.</p> <p>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.</p> <p>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>
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 <code>AccessPermissionChk</code> attribute is not 0.</p> <p>Type and dimension: string-scalar</p> <p>Example: "grp1"</p>

**Table 2-16** Optional attributes (*continued*)

Optional attribute	Description
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 2-16** Optional attributes (*continued*)

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 verification of the mount options.</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</p> <p>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</p> <p>Default: 0</p>
SnapUmount	<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>

**Table 2-16** Optional attributes (*continued*)

Optional attribute	Description
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 for Mount agent

The resource definition for this agent on HP-UX follows:

```

type Mount (
    static keylist RegList = { VxFSMountLock }
    static int IMF{} = { Mode = 3, MonitorFreq = 1,
        RegisterRetryLimit = 3 }
    static str IMFRegList[] = {MountPoint, BlockDevice, FSType }
    static keylist SupportedActions = { "mountpoint.vfd",
        "mounted.vfd", "mountentry.vfd", "vxfslic.vfd",
        chgmntlock }
    static str ArgList[] = { MountPoint, BlockDevice, FSType,
        MountOpt, FsckOpt, SnapUmount, CkptUmount, OptCheck,
        CreateMntPt, MntPtPermission, MntPtOwner, MntPtGroup,
        AccessPermissionChk, RecursiveMnt, VxFSMountLock, State }
    str MountPoint
    str BlockDevice
    str FSType
    str MountOpt
    str FsckOpt
    int SnapUmount
    int CkptUmount = 1
    int OptCheck = 0
    int CreateMntPt = 0
    int ReuseMntPt = 0
    str MntPtPermission
    str MntPtOwner

```

```
str MntPtGroup
int AccessPermissionChk = 0
boolean RecursiveMnt = 0
int VxFSMountLock = 1
static boolean AEPTimeout = 1
)
```

## Notes for Mount agent

The Mount agent has the following notes:

- [High availability fire drill](#)
- [VxFS file system lock](#)
- [IMF usage notes](#)
- [Support for loopback file system for HP-UX](#)
- [Enabling Level two monitoring for the Mount agent](#)

## High availability fire drill

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

For 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 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.

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

## 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 Level two monitoring for the Mount agent

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

To enable Level two monitoring, run the following commands:

- # haconf -makerw
- # hares -override resource\_name LevelTwoMonitorFreq
- # hares -modify resource\_name LevelTwoMonitorFreq 1
- # haconf -dump -makero

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

## Sample configurations for Mount agent

### Configuration 1 for Mount agent

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
```

### Configuration 2 for Mount agent

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 for Mount agent

In the following configuration, 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 for Mount agent

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 for Mount agent

The Mount agent uses the following debug log levels:

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

# Network agents

This chapter includes the following topics:

- [About the network agents](#)
- [IP agent](#)
- [NIC agent](#)
- [IPMultiNIC agent](#)
- [MultiNICA agent](#)
- [About the IPMultiNICB and MultiNICB agents](#)
- [IPMultiNICB agent](#)
- [MultiNICB agent](#)
- [DNS agent](#)

## About the network agents

Use network agents to provide high availability for networking resources.

All networking agents IP, NIC, IPMultiNIC, MultiNICA, IPMultiNICB and MultiNICB agents support IPv4 as well as IPv6 protocols.

## Agent comparisons

Agent comparisons may be made as described in the following sections.

### 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 SMIT. The physical interface is connected to a 802.1Q trunked port on the switch.

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

For example, create a 802.1Q interface called 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 NIC and IP agents, VCS supports Auto-port Aggregation (APA).

## High availability fire drill for IP agent

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:

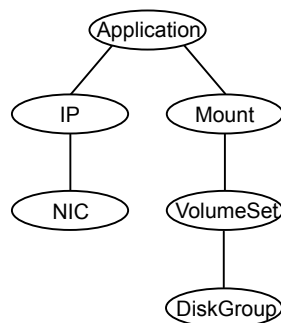
- Checks for the existence of a route to the IP from the specified NIC
- Checks for the existence of the interface configured in the IP resource

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

## Dependencies for IP agent

IP resources depend on NIC resources.

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



## Agent functions for IP agent

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	<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.</li><li>■ <code>device.vfd</code> Checks for the existence of the interface configured in the Device attribute.</li></ul>
Offline	Brings down the IP address that is specified in the Address attribute.
Monitor	Monitors the interface to test if the IP address that is associated with the interface is alive.
Clean	Brings down the IP address that is specified in the Address attribute.

## State definitions for IP agent

The state definitions for this agent follow:

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 for IP agent

For HP-UX:



**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>Example:</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>
One of the following attribute:	See <a href="#">Table 3-2</a> on page 81.
<ul style="list-style-type: none"> <li>■ NetMask</li> <li>■ PrefixLen</li> </ul>	

**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>
IfconfigTwice	<p>Causes an IP address to be configured twice using an ifconfig up-down-up 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>

Table 3-2 Optional attributes (*continued*)

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>
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>You must configure this attribute if the IP address is an IPv4 address.</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>For complete list of <code>ifconfig</code> options refer to <i>ifconfig manpage</i>.</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>

## Resource type definition for IP agent

The resource definition for this agent on HP-UX follows:

```
type IP (
static keylist RegList = { NetMask }
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 for IP agent

The sample configurations for this agent follow:

### Configuration 1

Configuration 1 for HP-UX follows:

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

### NetMask in decimal (base 10)

Configuration with decimal NetMask:

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

### NetMask in hexadecimal (base 16)

Configuration with hexadecimal NetMask:

```
IP ipres (
Device = lan0
```

```
Address = "192.203.47.61"  
NetMask = "0xfffff800"  
)
```

## IPv6 Configuration

```
IP ipres_v6 (  
Device = lan2  
Address = "fd4b:454e:205a:111:211:43ff:feaa:af70"  
PrefixLen = 64  
)
```

NIC Agent:

```
NIC nicres_v6 (  
Device = lan2  
NetworkHosts = { "fd4b:454e:205a:111:211:43ff:feaa:af71" }  
Protocol = IPv6  
)
```

## Debug log levels for IP agent

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 for NIC agent

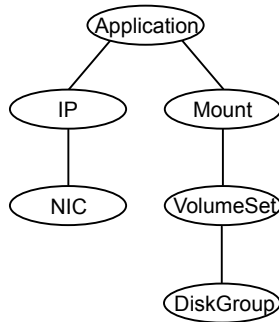
The high availability fire drill detects discrepancies between the VCS configuration and the underlying infrastructure on a node. These discrepancies might prevent a service group from going online on a specific node. For NIC resources, the high availability fire drill checks for the existence of the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

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

## Dependencies for NIC agent

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 a base IP address. The base 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 base IP addresses.

Before you use this agent:

- Verify that the NIC has the correct base 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 for NIC agent

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 for NIC agent

Monitor	<ul style="list-style-type: none"> <li>■ Tests the network card and network link. Pings the network hosts or broadcast address of the interface to generate traffic on the network. Counts the number of packets passing through the device before and after the address is pinged. If the count decreases or remains the same, the resource is marked FAULTED. If the NetworkHosts list is empty, or the ping test fails, the agent sends a ping to the device's broadcast address to generate network traffic. The agent checks for any response to the broadcast request. If there is no reply to the broadcast ping, the resource faults.</li> </ul>
Action	<ul style="list-style-type: none"> <li>■ device.vfd Checks for the existence of the interface configured in the Device attribute.</li> </ul>

## State definitions for NIC agent

The state definitions for this agent follow:

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 for NIC agent

**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"

**Table 3-3** Required attributes (*continued*)

Required attribute	Description
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> <p>When this agent is configured on IPv4, this attribute is optional.</p>

**Table 3-4** Optional attributes

Optional attribute	Description
NetworkType	<p>Type of network. VCS currently only supports Ethernet.</p> <p>Type and dimension: string-scalar</p> <p>Default: "ether"</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 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.</p> <p>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.</p> <p>Symantec recommends configuring at least one network host to report the correct NIC health status in case the network activity is low.</p> <p>Type and dimension: string-vector</p> <p>Example: "166.96.15.22" , "166.97.1.2"</p>

**Table 3-4** Optional attributes (*continued*)

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>

## Resource type definition for NIC agent

The resource definition for this agent on HP-UX follows:

```

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 for NIC agent

### Network Hosts for NIC agent

```

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

```



## IPv6 configuration for NIC agent

The following is a basic configuration for IPv6 with IP and NIC resources.

```
group nic_group (  
  SystemList = { sysA = 0, sysB = 1 }  
  Parallel = 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 for NIC agent

The NIC agent uses the following debug log levels:

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

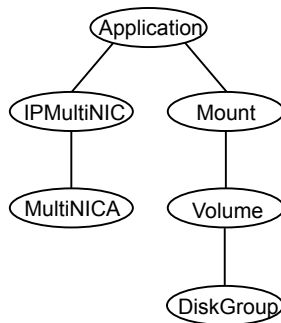
## IPMultiNIC agent

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 the virtual IP to a backup interface. If multiple service groups have IPMultiNIC resources associated with the same MultiNICA resource, only one group must have the MultiNICA resource. The other groups have Proxy resources pointing to it. You can use this agent for IP addresses on multiple-adapter systems.

## Dependencies for IPMultiNIC agent

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



## Agent functions for IPMultiNIC agent

Online	Configures a virtual IP address on the active interface of the MultiNICA resource. Also sends out a gratuitous ARP.
Offline	Removes the virtual IP address from the active 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 for IPMultiNIC agent

The state definitions for this agent follow:

ONLINE	Indicates that the specified IP address is assigned to one of the interfaces specified in the corresponding MultiNICA resource.
OFFLINE	Indicates that the specified IP address is not assigned to any interface of the MultiNICA resource.
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 for IPMultiNIC agent

**Table 3-5** Required attributes

Required attribute	Description
Address	Virtual IP address assigned to the active NIC. Type and dimension: string-scalar Example: <ul style="list-style-type: none"> <li>■ IPv4: "10.128.10.14"</li> <li>■ IPv6: "2001:DB8::"</li> </ul>
MultiNICResName	Name of the associated MultiNICA resource that determines the active NIC. Type and dimension: string-scalar Example: "mnic"
One of the two attributes: <ul style="list-style-type: none"> <li>■ NetMask</li> <li>■ PrefixLen</li> </ul>	See <a href="#">Table 3-6</a> on page 92.

**Table 3-6** Optional attributes

Optional attribute	Description
IfconfigTwice	<p>Causes an IP address to be configured twice using an <code>ifconfig</code> up-down-up 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>
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>You must configure this attribute if the IP address is an IPv4 address.</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> <p>Please refer to <code>ifconfig</code> man page for the list of supported Options.</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 for IPMultiNIC agent

The resource definition for this agent on HP-UX follows:

```

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

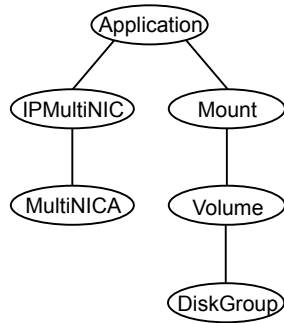
The MultiNICA agent represents a set of network interfaces and provides failover capabilities between them. You can use the agent to make IP addresses on multiple-adapter systems highly available or to monitor them. Each interface in a MultiNICA resource has a base IP address. You can use one base IP address for all interfaces, or you can specify a different IP address for use with each interface. 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. In the other service groups, configure Proxy resources that point to the MultiNICA resource in the first service group.

For more information on this agent, refer to See [“Notes for MultiNICA agent”](#) on page 98.

## Dependencies for MultiNICA agent

The MultiNICA resource does not depend on any other resources.

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

## Agent function for MultiNICA agent

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 for MultiNICA agent

The state definitions for this agent follow:

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 for MultiNICA agent

For HP-UX:

**Table 3-7** Required attributes

Required attribute	Description
Device	<p>List of interfaces and their base IP addresses.</p> <p>For each system you must localize the attribute with separate base IP addresses.</p> <p>Type and dimension: string-association</p> <p>Example: { lan0 = "192.205.8.42", lan3 = "192.205.8.42" }</p>
Protocol	For more information, see <a href="#">Table 3-8</a> .
One of the two attributes:	For more information, see <a href="#">Table 3-8</a> .
<ul style="list-style-type: none"> <li>■ NetMask</li> <li>■ PrefixLen</li> </ul>	

**Table 3-8** Optional attributes

Optional attribute	Description
ArpDelay	<p>Number of seconds to sleep between configuring an interface and sending out a broadcast to inform routers about the base IP address.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 1</p>
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 FAULTED. Increased values result in longer failover times, whether between the NICs or from system to system in the case of FAULTED NICs.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 20</p> <p>This value is the equivalent to two tries.</p>

**Table 3-8** Optional attributes (*continued*)

Optional attribute	Description
IfconfigTwice	<p>Causes an IP address to be configured twice, using an ifconfig up-down-up 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.</p> <p>Type and dimension: string-vector</p> <p>Example: "166.93.2.1", "166.97.1.2"</p>
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>You must configure this attribute if the IP address is an IPv4 address.</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 does not perform broadcast pings.</p> <p>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>



**Table 3-8** Optional attributes (*continued*)

Optional attribute	Description
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>
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 for MultiNICA agent

The resource definition for this agent on HP-UX follows:

```

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
)
```

## Notes for MultiNICA agent

- 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 interfaces 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 interface several times before it marks the resource OFFLINE. Engine logs record a detailed description of the events during a failover.
- The MultiNICA agent supports only one active interface on one IP subnet; the agent does not work with multiple active interfaces 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.
- Set the NIC here: `/etc/hostname.nic`
- Set the netmask here: `/etc/netmasks`

## Using RouteOptions for MultiNICA agent

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 for MultiNICA agent

The sample configurations for this agent agent follow:

### 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 interface 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, sysb }
    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, sysb }
    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 for MultiNICA agent

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 base IP addresses.
- All base 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 base 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 virtual IP address to the next active interface. You can use this agent for IP addresses on multiple-adapter systems.

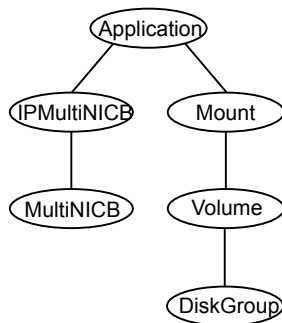
If multiple service groups have IPMultiNICB resources associated with the same MultiNICB resource, only one group should have a MultiNICB resource. The other groups should have a proxy resource pointing to the MultiNICB resource.

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

## Dependencies for IPMultiNICB agent

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 agent can control only one virtual IP address.

## Agent functions for IPMultiNICB agent

Online	Finds a working interface with the appropriate interface alias or interface name, and configures the virtual IP address on it.
Offline	Removes the virtual IP address.
Clean	Removes the virtual IP address.

Monitor	If the virtual 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 virtual 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 for IPMultiNICB agent

The state definitions for this agent follow:

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 virtual IP address could not be brought online, usually because all the interfaces configured in the MultiNICB resource have failed or the virtual IP address was removed out of VCS control.

## Attributes for IPMultiNICB agent

For HP-UX:

**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"

**Table 3-9** Required attributes (*continued*)

Required attribute	Description
BaseResName	<p>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.</p> <p>Type and dimension: string-scalar</p> <p>Example: "gnic_n"</p>
One of the two attributes: <ul style="list-style-type: none"> <li>■ NetMask</li> <li>■ PrefixLen</li> </ul>	<p>For more information, see See <a href="#">Table 3-10</a> on page 104.</p>

**Table 3-10** Optional attributes

Optional attribute	Description
DeviceChoice	<p>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.</p> <p>Type and dimension: string-scalar</p> <p>Default: 0</p> <p>Examples:</p> <p>DeviceChoice = "lan0"</p> <p>DeviceChoice = "1"</p>
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>This attribute is required if you configure this resource for IPv4 protocol.</p> <p>Type and dimension: string-scalar</p> <p>Example: "255.255.255.0"</p>



**Table 3-10** Optional attributes (*continued*)

Optional attribute	Description
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 for IPMultiNICB agent

The resource definition for this agent on HP-UX follows:

```

type IPMultiNICB (
  static keylist RegList = { Address, Options }
  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 for IPMultiNICB agent

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 for IPMultiNICB agent

The sample configurations for the IPMultiNICB and MultiNICB agent follows:

### Other sample configurations for IPMultiNICB and MultiNICB

Refer to the sample configurations in the MultiNICB agent.

## Debug log levels for IPMultiNICB agent

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 interfaces 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 interfaces before putting them under the agent's control. You must configure all the interfaces in a single MultiNICB resource with the base 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 for MultiNICB agent

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 for MultiNICB agent

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

Some examples of the command syntax are as follows:

Link test only on interface 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
```

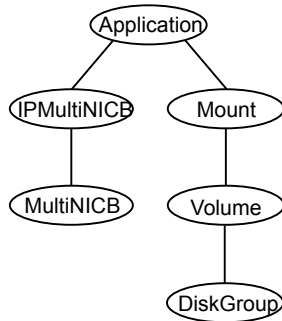
## Auto Port Aggregation (APA) support for MultiNICB agent

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 for MultiNICB agent

The MultiNICB resource does not depend on any other resources.

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

## Agent functions for MultiNICB agent

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 for MultiNICB agent

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 for MultiNICB agent

For HP-UX:

**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	Required to use the IPv6 protocol.
NetworkHosts	Required to use the IPv6 protocol.

**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>
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>

**Table 3-12** Optional attributes (*continued*)

Optional attribute	Description
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 (*continued*)

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 Corporation 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 (*continued*)

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 for MultiNICB agent

The resource definition for this agent on HP-UX follows:

```

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 for MultiNICB agent

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 `mnicb_postchange` trigger is provided with the agent. You can customize this sample script as needed or write one from scratch.

The sample script does the following:

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

```
MultiNICB: Device lan0 status changed from down to up.
```

## IPMultiNICB and MultiNICB configuration for MultiNICB agent

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 for MultiNICB agent

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)
- Alias to hostname or 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 if the Resource Records need to be dynamically added and deleted from the DNS servers during failover. The agent updates the name server

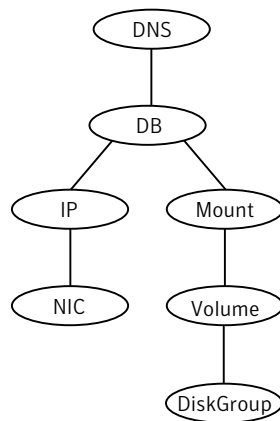
with the new resource record mappings while failing over and allows the clients to connect to the failed over instance of the application.

For important information about this agent, refer to [Agent notes for DNS agent](#)

## Dependencies for DNS agent

No dependencies exist for the DNS resource.

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



## Agent functions for DNS agent

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> <p><b>Note:</b> The DNS agent does not send any update for a resource record if it is already present on the name server.</p>
--------	---

Offline	<p>Removes the Online lock file.</p> <p>If attribute OffDeIrr 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> <p>The monitor entry point also sends periodic refresh requests to DNS server if the RefreshInterval attribute is set.</p>
Clean	<p>Removes the Online lock file, if it exists. If attribute OffDeIrr is true, clean removes all records that the ResRecord keys define.</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 for DNS agent

The state definitions for this agent follow:

ONLINE	<p>Online lock file exists and at least one name server can return all configured resource records.</p>
OFFLINE	<p>At least one of the following is true:</p> <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 Indicates that the DNS resource is not configured correctly. 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 for DNS agent

**Table 3-13** Required attributes

Required attribute	Description
Domain	<p>A string representing the DNS zone that the agent administers.</p> <p>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 (*continued*)

Required attribute	Description
ResRecord	

**Table 3-13** Required attributes (*continued*)

Required attribute	Description
	<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. The A type ResRecord configuration should be as follows: <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> <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 types are as</p>



**Table 3-13** Required attributes (*continued*)

Required attribute	Description
	<p>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 an IPv4 address (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 the agent logs a warning message to the engine log file. This 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 3-14** Optional attributes

Optional attribute	Description
TTL	<p>This attribute (a non-zero integer) represents the Time To Live (TTL) 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 TTL 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 3-14** Optional attributes (*continued*)

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. This attribute should be configured only when the DNS server configured is a Unix based DNS server.</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 functions 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 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 and clean agent functions to remove all records that the ResRecord key defines. You must set the value of this attribute to 1 (true) to have the agent remove all the records.</p> <p>Type and dimension: boolean-scalar</p> <p>Default: 0</p> <p>Example: 1</p>

**Table 3-14** Optional attributes (*continued*)

Optional attribute	Description
UseGSSAPI	<p>Use the UseGSSAPI attribute if the DNS server that you have configured is a Windows DNS server and only if it accepts secure dynamic updates.</p> <p><b>Note:</b> Do not set this attribute if the Windows DNS server accepts non-secure updates.</p> <p>If this attribute is set to 1, the agent uses the -g option with the nsupdate command.</p> <p>See <a href="#">"Agent notes for DNS agent"</a> on page 124. for more information on requirements to use the DNS agent with the secure Windows DNS server.</p> <p>Type and dimension: boolean-scalar</p> <p>Default: 0</p> <p>Example: 1</p>
RefreshInterval	<p>This attribute represents the time interval in seconds after which the DNS agent attempts to refresh the resource records (RRs) on the DNS servers. The default value of zero indicates that the DNS agent does not attempt to refresh the records on the DNS servers. The DNS agent writes the warning message to the logs if it is not able to refresh the DNS records.</p> <p><b>Note:</b> The refresh request is sent in the next monitor cycle after the RefreshInterval period is reached.</p> <p>If the DNS agent is unable to refresh the DNS records, and the records are removed as a result of a scavenging operation or by the DNS administrator, the DNS resource will fault.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 0</p> <p>Example: 3600</p>

**Table 3-14** Optional attributes (*continued*)

Optional attribute	Description
CleanRRKeys	<p>Use this attribute to direct the online agent function to clean up all the existing DNS records for the configured keys before adding new records. The default value (0) disables this behavior.</p> <p><b>Note:</b> If multiple DNS resources are configured with the same key value in their ResRecord attribute, then do not set this attribute value to 1.</p> <p>Type and dimension: boolean-scalar</p> <p>Default: 0</p> <p>Example: 1</p>

## Resource type definition for DNS agent

The resource definition for this agent on HP-UX follows:

```

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

```

## Agent notes for DNS agent

The DNS agent has the following notes:

- [About using the VCS DNS agent on UNIX with a secure Windows DNS server](#)
- [High availability fire drill for DNS agent](#)

- [Monitor scenarios for DNS agent](#)
- [Sample Web server configuration for DNS agent](#)
- [Secure DNS update for BIND 9 for DNS agent](#)
- [Setting up secure updates using TSIG keys for BIND 9 for DNS agent](#)

## About using the VCS DNS agent on UNIX with a secure Windows DNS server

This section describes the requirements for using the DNS agent with a secure Windows DNS server. Note that there are no special requirements for sending non-secure updates to a Windows DNS server.

### Software requirement for DNS agent

For the secure updates on Windows DNS server to work, the VCS DNS agent on UNIX requires BIND version 9.7.2-P3 or later installed on all cluster nodes.

### Configuration requirement for DNS agent

The VCS DNS agent on UNIX requires setting up Kerberos authentication with the Windows DNS server and configuring the domain and DNS server information in `/etc/resolv.conf` at the client node.

To set up the Kerberos authentication from the UNIX host to the Windows DNS server, configure the Kerberos configuration file (`/etc/krb5.conf` or `/etc/krb/krb5.conf`) to use the Windows DNS server as Key Distribution Centre (KDC).

A sample Kerberos configuration file with domain `privdns.sym` and DNS server `master.privdns.sym` is as follows:

```
[libdefaults]
default_realm = PRIVDNS.SYM
dns_lookup_realm = true
dns_lookup_kdc = true
default_tkt_enctypes = des-cbc-md5
default_tgs_enctypes = des-cbc-md5
ticket_lifetime = 24h
renew_lifetime = 7d
forwardable = true
allow_weak_crypto = true
[realms]
PRIVDNS.SYM = {
kdc = master.privdns.sym:88
kpasswd_server = master.privdns.sym:464
admin_server = master.privdns.sym
```

```
}  
[domain_realm]  
.privdns.sym = PRIVDNS.SYM  
privdns.sym = PRIVDNS.SYM
```

---

**Note:** Symantec does not support KDC and Domain Controller/DNS located on different servers.

---

Authenticate all the nodes on the cluster (on which the DNS agent is configured to run) with the Active directory. Use kinit on your user account and use klist to verify that you have a ticket to the configured realm principal. Refer to the man page of kinit for more information on obtaining Kerberos ticket granting tickets from KDC.

---

**Note:** The DNS agent requires a node to be authenticated with Kerberos all the time. Renew the obtained tickets periodically if your authentication method requires you to do so.

---

A sample run of kinit and klist for the above configuration with user vcsdns will look as follows:

```
# kinit vcsdns  
Password for vcsdns@PRIVDNS.SYM:  
# klist  
Ticket cache: FILE:/tmp/krb5cc_0  
Default principal: vcsdns@PRIVDNS.SYM  
Valid starting Expires Service principal  
12/14/09 16:17:37 12/15/09 02:19:09 krbtgt/PRIVDNS.SYM@PRIVDNS.SYM  
renew until 12/21/09 16:17:37
```

If the environment variable KRB5CCNAME is set to some non-default location (default is /tmp), then VCS will not inherit it by default and will look for the Kerberos tickets in default location /tmp.

To resolve this issue, un-set the environment variable KRB5CCNAME and run the kinit command again. This will update the Kerberos tickets in default location (/tmp). Else, for a customized location (for example, /cache/krb\_ticket) for Kerberos tickets, add an entry in opt/VRTSvcs/bin/vcsenv file on each cluster node before VCS starts:

```
KRB5CCNAME="FILE:/cache/krb_ticket"  
export KRB5CCNAME
```

Update `/etc/resolv.conf` on your client node to add information for the Windows DNS server and the configured domain.

## High availability fire drill for DNS agent

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 for DNS agent

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 3-15](#) summarizes the monitor scenarios for the Online lock files.

**Table 3-15** 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 for DNS agent

Take the example of a Web server. A browser requests the URL `http://www.example.com` that maps to the canonical name `server1.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 server one to server two (`server2.example.com`), the domain name servers need a new canonical name

mapping for `www.example.com`. After the failover, the DNS resource updates this mapping of `www.example.com` to point to canonical name `server2.example.com`

---

**Note:** In this configuration, the Domain attribute should be configured with value `"example.com"`

---

## Secure DNS update for BIND 9 for DNS agent

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 BIND 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 for DNS agent

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

To use secure updates using TSIG keys, perform the following steps at the DNS server:

- 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 HOST example.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-update` 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 are 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 for DNS agent

This section contains sample configurations for this agent.

### Basic IPv6 configuration for DNS agent

This sample configuration provides basic configuration for IPv6 support. In the following configuration, `nic_value` represents the base NIC value for the platform

For example: `lan0`

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

```
DNS ipv6group_dns_res (
    Critical = 0
    Domain = "example.com"
    TSIGKeyFile = "/var/tsig/Kipv6.vcsd.net.+157+18435.private"
    StealthMasters = { "2001:db8:c18:2:69c4:3251:bac1:6cbe" }
    ResRecord = {
        vcssystemCv6 = "2001:db8:c18:2:214:4fff:fe96:8833",
        sysC = vcssystemCv6 }
    )

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 for DNS agent

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 = { www = server1 }
    ResRecord @sysB = { www = server2 }
```

```
        OffDelRR = 1
    )
```

## IPv4 A sample configuration for DNS agent

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 for DNS agent

The DNS agent uses the following debug log levels:

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

# File share agents

This chapter includes the following topics:

- [About the file service agents](#)
- [NFS agent](#)
- [NFSRestart agent](#)
- [Share agent](#)
- [About the Samba agents](#)
- [SambaServer agent](#)
- [SambaShare agent](#)
- [NetBios agent](#)

## 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.

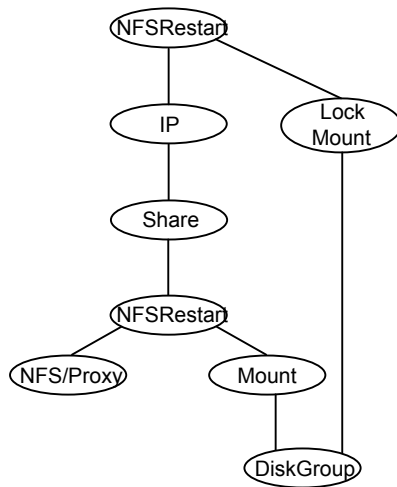
You should configure only a single NFS resource in a service group on a node. If you have more than one service group that uses the NFS resource, the other service groups must use a Proxy resource. The Proxy resource can point to the NFS resource in the first group. Duplicate NFS resources will cause 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 NFS agent

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 for NFS agent

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 <code>nfsd</code> daemons, and versions 1 and 3 of the <code>rpc.mountd</code> 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 <code>nfsd</code> and <code>rpc.mountd</code> daemons.

## State definitions for NFS agent

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 for NFS agent

**Table 4-1** Optional attributes for HP-UX

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

```

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 for NFS agent

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 for NFS agent

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 [Notes for NFSRestart agent](#)

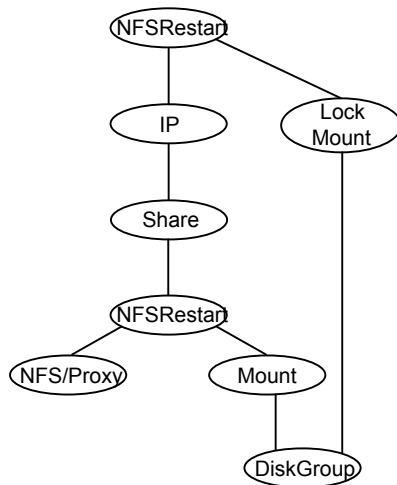
## Dependencies for NFSRestart agent

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. Both the NFSRestart resources provide combined protection from potential corruption of NFS locks and

potential NFS ACK storms. 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 for NFSRestart agent

The agent functions for this agent follow:

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	<p>For the lower NFSRestart resource:</p> <ul style="list-style-type: none"><li>■ The monitor agent function does nothing.</li></ul> <p>For the upper NFSRestart resource:</p> <ul style="list-style-type: none"><li>■ If the value of the NFSLockFailover attribute is 1, the agent monitors smsyncd daemon. It restarts the smsyncd daemon if it is not running.</li><li>■ Monitors the statd and lockd daemons</li></ul>
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 for NFSRestart agent

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 for NFSRestart agent

**Table 4-2** Required attributes

Required attribute	Description
NFSRes	Name of the NFS resource. 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 for NFSRestart agent

```
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
)
```

## Notes for NFSRestart agent

The NFSRestart agent has the following notes:

- [About high availability fire drill](#)
- [Providing a fully qualified host name](#)

### 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 for NFSRestart agent

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 for NFSRestart agent

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.

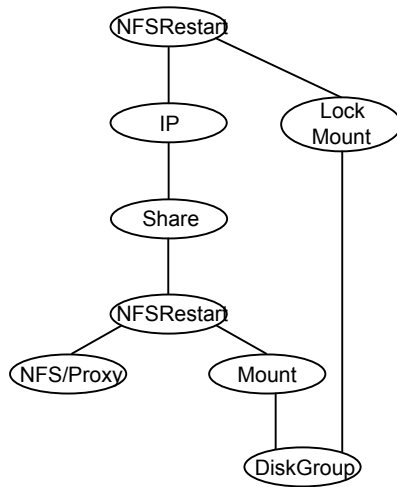
For important information on this agent, refer to:

[Notes for Share agent](#)

## Dependencies for Share agent

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 for Share agent

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.

## State definitions for Share agent

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 specified directory is unshared outside the control of VCS.

## Attributes for Share agent

**Table 4-4** Required attributes

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

**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 for Share agent

```
type Share (
  static str ArgList[] = { PathName, Options }
  static int NumThreads = 1
  str PathName
  str Options
)
```

## Notes for Share agent

The following section contains notes on the Share agent.

- [High availability fire drill](#)

### 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 for Share agent

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 for Share agent

The Share agent uses the following debug log levels:

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 `sm bd` 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 for the Samba agents

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

```
# smb -V
```

## 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:

```
# smb -b | grep CONFIGFILE
```

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

- ◆ Enter the following command:

```
# smb -b | grep PIDDIR
```

## SambaServer agent

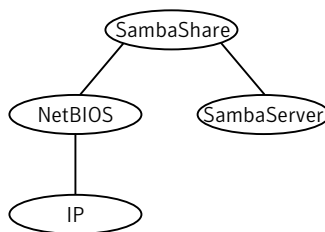
The SambaServer agent starts, stops, and monitors the `smbd` process as a daemon. Only one resource of this type is permitted. You can use the agent to make a `smbd` daemon highly available.

The `smbd` daemon provides Samba share services. The agent verifies that Samba is running by reading the `pid` of `smbd` 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 for SambaServer agent

No dependencies exist for the SambaServer resource.

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



## Agent functions for SambaServer agent

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 forcefully if required.

## State definitions for SambaServer agent

ONLINE	Indicates that the <code>smbd</code> 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 for SambaServer agent

**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 (*continued*)

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>
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 attribute 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: <code>"/var/opt/samba/locks/smbd.pid"</code></p>
SambaTopDir	<p>Parent path of Samba daemon and binaries.</p> <p>Type and dimension: string-scalar</p> <p>Example: <code>"/opt/samba"</code></p> <p>SambaServer agent uses SambaTopDir attribute value in an open entry point to determine the complete path of samba executables. If this attribute is configured after the resource is enabled, please disable and enable the resource again to bring this into effect as follows:</p> <pre># hares -modify &lt;res&gt; Enabled 0 # hares -modify &lt;res&gt; Enabled 1</pre>
SocketAddress	<p>The IP address where the Samba daemon (smbd) listens for connections. Configure the SocketAddress attribute if you are configuring multiple SambaServer resources on a node.</p> <p><b>Note:</b> Only IPv4 addresses are supported.</p> <p>Type and Dimension: string-scalar</p> <p>Example: <code>"10.128.10.14"</code></p>

## Resource type definitions for SambaServer agent

```
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 for SambaServer agent

The sample configurations for this agent follow:

```
SambaServer Samba_SambaServer (  
ConfFile = "/etc/opt/samba/smb.conf"  
SambaTopDir = "/opt/samba"  
LockDir = "/var/opt/samba/locks"  
)
```

## Debug log levels for SambaServer agent

The SambaServer agent uses the following debug log levels:

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

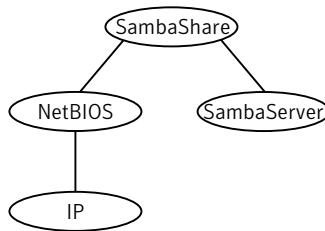
## SambaShare agent

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 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 for SambaShare agent

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

**Figure 4-5** Sample service group for a SambaShare resource

## Agent functions for SambaShare agent

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 connections with the particular samba share, removes its entry from the samba configuration file and reloads the configuration.

## State definitions for SambaShare agent

ONLINE	Indicates that the share is available.
OFFLINE	Indicates that the share is not available.
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 for SambaShare agent

**Table 4-8** Required attributes

Required attribute	Description
SambaServerRes	Name of the SambaServer resource. Type and dimension: string-scalar Example: "smb_res1"

**Table 4-8** Required attributes (*continued*)

Required attribute	Description
ShareName	Name of the share resource as exported by samba. <b>Note:</b> This name can be different from the SambaShare resource name. 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 for SambaShare agent

```

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 for SambaShare agent

```

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

```

## Debug log levels for SambaShare agent

The SambaShare agent uses the following debug log levels:

DBG\_1, DBG\_3, DBG\_5

# NetBios agent

The NetBios agent starts, stops, and monitors the nmbd daemon. Only one resource of this type is permitted. 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.

---

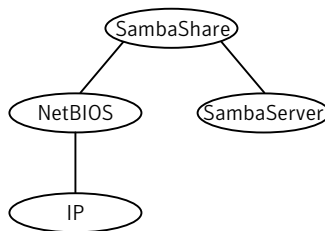
**Note:** The nmbd broadcasts the NetBIOS name, or the name by which the Samba server is known in the network.

---

## Dependencies for NetBios agent

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 for NetBios agent

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. Also verifies that the nmbd daemon is running by reading its pid file.



Clean	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, forcibly when necessary.
-------	--

## State definitions for NetBios agent

ONLINE	Indicates that the specified NetBIOS name and 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 any 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 resource has become offline unexpectedly outside of VCS control.

## Attributes for NetBios agent

**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 Example: "samba_demon" <b>Note:</b> Samba has a limitation of 15 characters for NetBios names and aliases.
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> 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> <p><b>Note:</b> Samba has a limitation of 15 characters for NetBios names and aliases.</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 set to 1, the agent sets Samba as Domain Master. Note that there can be only one domain master in a domain.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 0</p>
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 nonstandard configuration file name or path. If this attribute 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: "/var/opt/samba/locks/nmbd.pid"</p>

## Resource type definition for NetBios agent

```
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 for NetBios agent

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

## Debug log levels for NetBios agent

The NetBios agent uses the following debug log levels:

DBG\_1, DBG\_5

# Service and application agents

This chapter includes the following topics:

- [About the services and applications agents](#)
- [Apache HTTP server agent](#)
- [Application agent](#)
- [CoordPoint agent](#)
- [Process agent](#)
- [ProcessOnOnly agent](#)
- [HPVirtualMachine agent](#)
- [HPVSwitch agent](#)

## About the services and applications agents

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

## Apache HTTP server agent

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

This agent supports the Apache HTTP server 1.3, 2.0, and 2.2. It also supports the IBM HTTP Server 1.3, 2.0 and 7.0.0.0.

This agent can detect when an Apache HTTP 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.

---

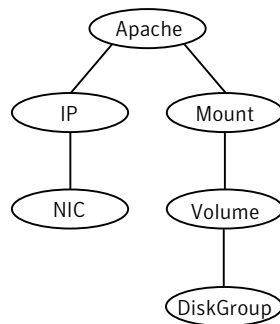
For more information regarding this agent:

See [“Apache HTTP server notes”](#) on page 163.

## Dependencies

This type of resource depends on IP and Mount resources.

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



## Agent functions

Online

To start the Apache HTTP server, the agent:

- Executes the httpdDir/httpd program with the appropriate arguments if the httpdDir program specifies the full path of the directory in which the httpd binary file is located.
- Alternatively, if the httpdDir attribute specifies the full path of the Apache HTTP server binary file, the binary file is executed with appropriate arguments.

When you specify a file with the EnvFile attribute, the file is sourced before the agent executes the Apache HTTP server commands.

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, if httpdDir specifies the full path of the directory in which the httpd binary file is located.</li><li>■ Alternatively, if the httpdDir attribute is used to specify the full path of the Apache HTTP server binary, the binary file is executed with appropriate arguments.</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 Apache HTTP server commands.</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.</p> <p>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 HTTP 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	Full path and file name of the main configuration file for the Apache server. Type and dimension: string-scalar Example: "/apache/server1/conf/httpd.conf"
httpdDir	Full path of the Apache HTTP server binary file or full path of the directory in which the httpd binary file is located. Type and dimension: string-scalar Example: "/apache/server1/bin"
PidFile	This attribute is required when you want to enable the detection of a graceful shutdown outside of VCS control. See <a href="#">Table 5-2</a> on page 159.

**Table 5-2** Optional attributes

Optional attribute	Description
DirectiveAfter	A list of directives that httpd processes after reading the configuration file. Type and dimension: string-association Example: DirectiveAfter{} = { KeepAlive=On }
DirectiveBefore	A list of directives that httpd processes before it reads the configuration file. Type and dimension: string-association Example: DirectiveBefore{} = { User=nobody, Group=nobody }
User	Account name the agent uses to execute the httpd program. If you do not specify this value, the agent executes httpd as the root user. Type and dimension: string-scalar Example: "apache1"

Table 5-2 Optional attributes (continued)

Optional attribute	Description
EnableSSL	<p>If this attribute is set to 1 (true) the online agent function will add support for SSL, by including the option <code>-DSSL</code> in the start command.</p> <p>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>If this attribute is set to 0 (false) the agent excludes the SSL support.</p> <p>Type and dimension: boolean-scalar</p> <p>Default: 0</p> <p>Example: "1"</p>
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 <code>HostName</code> attribute is only required when the value of <code>SecondLevelMonitor</code> 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 <code>SecondLevelMonitor</code> 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 Apache HTTP server commands. With Apache 2.0, the file <code>ServerRoot/bin/envvars</code>, which is supplied in most Apache 2.0 distributions, is commonly used to set the environment before executing <code>httpd</code>. Specifying this attribute is optional. If <code>EnvFile</code> is specified, the shell for user must be Bourne, Korn, or C shell.</p> <p>Type and dimension: string-scalar</p> <p>Example: "/apache/server1/bin/envvars"</p>



**Table 5-2** Optional attributes (*continued*)

Optional attribute	Description
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 Apache HTTP server. For the agent to detect the graceful shutdown of the Apache HTTP server, the value of the IntentionalOffline resource type attribute must be 1 (true).</p> <p>Type and dimension: string-scalar</p> <p>Example: <code>/var/run/httpd.pid</code></p>
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 httpd program. Refer to the httpd man pages for more information about the <code>-R</code> option.</p> <p>Type and dimension: boolean-scalar</p> <p>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).</p> <p>Type and dimension: boolean-scalar</p> <p>Default: 0</p> <p>Example: <code>"1"</code></p>

**Table 5-2** Optional attributes (*continued*)

Optional attribute	Description
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 SecondLevelTimeout window expires, the monitor agent function no longer blocks on the program sub-process. It does, however, report that the resource is offline. The value should be high enough to allow the second level monitor enough time to complete. The value should be less than the value of the agent's MonitorTimeout.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 30 Table</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>

**Table 5-3** Resource type attribute

Optional 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
int SecondLevelTimeout = 30
str ConfigFile = 0
str DirectiveAfter{}
str DirectiveBefore{}

static boolean IntentionalOffline = 0
)
```

## Apache HTTP server notes

The Apache Apache HTTP server has the following notes:

- See [“Tasks to perform before you use the Apache HTTP server agent”](#) on page 163.
- See [“About detecting application failure”](#) on page 164.
- See [“About bringing an Apache HTTP server online outside of VCS control”](#) on page 164.
- See [“About high Availability fire drill”](#) on page 165.

## Tasks to perform before you use the Apache HTTP 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 HTTP 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 HTTP server online outside of VCS control”](#) on page 164.
- 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.
- 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.

If the binary file `ab` is not found, Apache agent uses the `ab2` binary file for detail monitoring.

## About bringing an Apache HTTP server online outside of VCS control

When you bring an Apache HTTP 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 HTTP server outside of VCS control

- 1 Source the environment file if required.
- 2 Start the Apache HTTP 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
```

---

**Note:** You can specify the full path of a binary file without having `httpd` as part of `httpdDir` attribute.

For example:

---

```
/usr/sbin/apache2 -f /etc/httpd/conf/httpd.conf -k start
```

## 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 HTTP 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 (
    httpdDir = "/apache/bin"
    HostName = vcshp1
    Port = 8888
    User = root
    SecondLevelMonitor = 1
    ConfigFile = "/apache/conf/httpd.conf"
)

DiskGroup Apache_dg (
    DiskGroup = apc1
)

IP Apache_ip (
    Device = lan0
    Address = "11.123.99.168"
    NetMask = "255.255.254.0"
)

Mount Apache_mnt (
    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 can be on local storage or shared storage. 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*.

Prevention Of Concurrency Violation (ProPCV), a group level attribute, can be enabled through the Command Line Interface (CLI) to prevent an online resource on a node from coming online on another node, outside of VCS control, in the same cluster. This action prevents data corruption of resources. The attribute can only be set for a local failover type group. For more information about ProPCV action entry point, refer to the *Veritas Cluster Server Administrator's Guide*.

## High availability fire drill for Application agent

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:

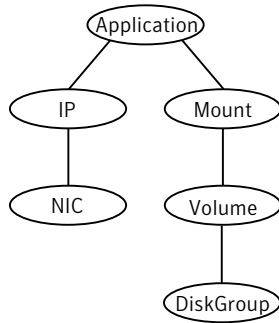
- The availability of the specified program (program.vfd)
- Execution permissions for the specified program (program.vfd)
- The existence of the specified user on the host (user.vfd)
- The existence of the same binary on all nodes (cksum.vfd)

For more information refer to the *Veritas Cluster Server Administrator's Guide*.

## Dependencies for Application agent

No fixed dependency exists for Application agent.

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 for Application agent

Online	<p>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.</p> <p>To bring the resource online, the agent function performs the command:</p> <pre>su [-] user -c command_to_online_resource</pre>
Offline	<p>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.</p> <p>To take the resource offline, the agent function performs the command:</p> <pre>su [-] user -c command_to_offline_resource</pre>



Monitor	<p>If you specify the <code>MonitorProgram</code> attribute, the agent executes the user defined <code>MonitorProgram</code> in the user-specified context. If you specify the <code>PidFiles</code> attribute, the routine verifies that the process ID that is found in each listed file is running. If you specify the <code>MonitorProcesses</code> attribute, the routine verifies that each listed process is running in the context of the user you specify.</p> <p>Use any combination among these attributes (<code>MonitorProgram</code>, <code>PidFiles</code>, or <code>MonitorProcesses</code>) to monitor the application.</p> <p>If any of the processes that are specified in either <code>PidFiles</code> or <code>MonitorProcesses</code> is determined not to be running, the monitor returns <code>OFFLINE</code>. If the process terminates ungracefully, the monitor returns <code>OFFLINE</code> and failover occurs.</p> <p>To monitor the resource, the agent function performs the command:</p> <pre>su [-] user -c <i>command_to_monitor_resource</i></pre>
<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>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.</p>
<code>imf_register</code>	<p>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 <code>PidFiles</code> and the <code>MonitorProcesses</code> attribute.</p>
Clean	<p>Terminates processes specified in <code>PidFiles</code> or <code>MonitorProcesses</code>. Ensures that only those processes (that are specified in the <code>MonitorProcesses</code> attribute) running with the user ID specified in the <code>User</code> attribute are killed. If the <code>CleanProgram</code> is defined, the agent executes the <code>CleanProgram</code>.</p> <p>To forcefully stop the resource, the agent function performs the command:</p> <pre>su [-] user -c <i>command_to_clean_resource</i></pre> <p>Note that the agent uses the <code>su -</code> option only when the attribute <code>UseSUDash</code> is enabled (1). The <code>UseSUDash</code> attribute is disabled (0) by default.</p>

Action	<p>The various functions of the action entry point are as follows:</p> <ul style="list-style-type: none"><li>■ <code>program.vfd</code> Checks the availability of the specified program and the execution permissions for the specified program.</li><li>■ <code>user.vfd</code> Checks the existence of the specified user on the host.</li><li>■ <code>cksum.vfd</code> Checks the existence of the same binary on all nodes.</li><li>■ <code>propcv</code> Invokes the AMF call with arguments to decide whether to allow or prevent processes from starting for an application resource, outside the VCS control, in the cluster.</li><li>■ <code>getcksum</code> Returns the checksum of the specified program</li></ul>
--------	--

## State definitions for Application agent

ONLINE	Indicates that all processes that are specified in the <code>PidFiles</code> and the <code>MonitorProcesses</code> attribute are running and that the <code>MonitorProgram</code> returns ONLINE.
OFFLINE	Indicates that at least one process that is specified in the <code>PidFiles</code> attribute or <code>MonitorProcesses</code> is not running, or that the <code>MonitorProgram</code> returns OFFLINE.
UNKNOWN	Indicates an indeterminable application state or invalid configuration.
FAULTED	Indicates that the process has terminated unexpectedly or <code>MonitorProgram</code> returns "offline" unexpectedly.

## Attributes for Application agent

**Table 5-4** Required attributes for HP-UX

Required attribute	Description
StartProgram	<p>The executable, 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. This executable can be on local storage or shared storage.</p> <p>For example, if the attribute for StartProgram is</p> <pre>/usr/sbin/vxnotify -g dg00 -m &gt;&gt; /var/log/vxnotify.log</pre> <p>(and vxnotify is blocking command) set it like:</p> <pre>/usr/sbin/vxnotify -g dg00 -m &gt;&gt; /var/log/vxnotify.log &amp;</pre> <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, 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. This executable can be on local storage or shared storage..</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>
<p>At least one of the following attributes:</p> <ul style="list-style-type: none"> <li>■ MonitorProcesses</li> <li>■ MonitorProgram</li> <li>■ PidFiles</li> </ul>	<p>See <a href="#">Table 5-5</a> on page 172.</p>

**Table 5-5** Optional attributes for HP-UX

Optional attribute	Description
CleanProgram	<p>The executable, 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. This executable can be on local storage or shared storage.</p> <p><b>Note:</b> Symantec recommends to have the CleanProgram on the local storage so that in case of loss of storage connectivity VCS can take appropriate action to stop the application.</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 to be 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, 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. This executable can be on local storage or shared storage.</p> <p>MonitorProgram can return the following VCSAgResState values: OFFLINE value is 100 or 1; online values range from 101 to 110 or 0 (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>If MonitorProgram is configured and not available, then resource state will be:</p> <ul style="list-style-type: none"> <li>■ OFFLINE if the resource was in OFFLINE state and not waiting for any action .</li> <li>■ UNKNOWN if the resource was in any other state or waiting for some action.</li> </ul> <p>Type and dimension: string-scalar</p> <p>Example: "/usr/sbin/sample_app_monitor all"</p>

**Table 5-5** Optional attributes for HP-UX (*continued*)

Optional attribute	Description
PidFiles	<p>A list of PID (process ID) files that contain the PID 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 ID 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><b>Note:</b> If configured user does not exist then the resource state will be UNKNOWN.</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><b>Note:</b> Please make sure that the EnvFile adheres the default shell syntax of the configured user.</p> <p>Type and dimension: string-scalar</p> <p>Default: ""</p> <p>Example: /home/username/envfile</p>

**Table 5-5** Optional attributes for HP-UX (*continued*)

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 for Application agent

```

type Application (
    static keylist SupportedActions = { "program.vfd",
        "user.vfd", "cksum.vfd", getcksum, propcv }
    static int IMF{} = { Mode = 3, MonitorFreq = 1,
        RegisterRetryLimit = 3 }
    static keylist IMFRegList = { MonitorProcesses, User,
        PidFiles, MonitorProgram }
    static keylist RegList = { MonitorProcesses, User }
    static str ArgList[] = { State, IState, User,
        StartProgram, StopProgram, CleanProgram, MonitorProgram,
        PidFiles, MonitorProcesses, EnvFile, UseSUDash }
    str User = root
    str StartProgram
    str StopProgram
    str CleanProgram
    str MonitorProgram
    str PidFiles[]
    str MonitorProcesses[]
    str EnvFile
    boolean UseSUDash
)

```

## Notes for Application agent

### 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 the following table:

Table 5-6

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.

---

**Note:** For a resource, if a PID file configured in the PidFiles attribute and a process configured in the MonitorProcesses attribute have the same process ID (PID), then the resource fails to register to IMF.

---

## Requirement for programs

The programs specified in StartProgram, StopProgram, MonitorProgram, CleanProgram should not continuously write to STDOUT or STDERR. If required, please redirect STDOUT and STDERR to some file.

## Requirement for default profile

The default profile of configured user should not have any blocking command such as `bash` or any other command such as `exec` that changes the behavior of the shell. This may lead to unexpected behavior.

## Sample configurations for Application agent

The sample configurations for this agent follow:

### Sample Configuration 1 for Application agent

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 for Application agent

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 for Application agent

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:

See [“Notes for the CoordPoint agent”](#) on page 179.

## Dependencies for CoordPoint agent

No dependencies exist for the CoordPoint resource.

## Agent functions for CoordPoint agent

Monitor	<p>Enables the CoordPoint agent to validate the node registrations in the coordination points and confirms that the coordination points are accessible.</p> <p>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 <code>hastatus -sum</code>.</p> <p>The CoordPoint agent also performs I/O fencing reporting activities.</p> <p>See <a href="#">“CoordPoint agent I/O fencing reporting activities”</a> on page 179.</p>
---------	---

## State definitions for CoordPoint agent

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 for CoordPoint agent

**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</p> <p>Default: "0"</p>

## Resource type definition for CoordPoint agent

```
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 for CoordPoint agent

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 for CoordPoint agent

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 does an exact match of configured PathName and Arguments with the processes in the process table. To clarify, the processes that `ps` command lists should have exact match of PathName and Arguments for all the configured 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 for Process agent

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:

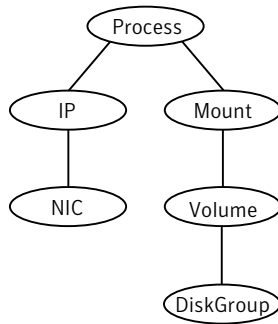
- The existence of a binary executable for the specified process (`program.vfd`)
- The existence of the same binary on all nodes (`program.vfd`)

For more information refer to the *Veritas Cluster Server Administrator's Guide*.

## Dependencies for Process agent

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 for Process agent

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.
Clean	Terminates all ongoing resource actions and takes the resource offline, forcibly when necessary.
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).

## State definitions for Process agent

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 along with the arguments.
OFFLINE	Indicates that the specified process is not running.
FAULTED	Indicates that the process has terminated unexpectedly.
UNKNOWN	Indicates that the agent can not determine the state of the process.

## Attributes for Process agent

**Table 5-8** Required attribute for HP-UX

Required attribute	Description
PathName	<p>Absolute path 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 for HP-UX

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>
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 for Process agent

```

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  
)
```

## Usage notes for Process agent

The Process agent has the following notes:

- [Requirement for programs](#)

### Requirement for programs

The programs specified in PathName should not continuously write to STDOUT or STDERR. If required, please redirect STDOUT and STDERR to some file.

## Sample configurations for Process agent

### Configuration 1 for Process agent

Configuration 1 for HP-UX follows:

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

### Configuration 2 for Process agent

Configuration 2 for HP-UX follows:

```
cluster vcs_test (  
    UserNames = { admin = bIJbIDiFJeJJhRJdIG }  
    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 for Process agent

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 or to monitor it. 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 along with the arguments.
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 for HP-UX

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 for HP-UX

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
)

```

## ProcessOnOnly agent usage notes

The ProcessOnOnly agent has the following notes:

- [Requirement for programs](#)

### Requirement for programs

The programs specified in PathName should not continuously write to STDOUT or STDERR. If required, please redirect STDOUT and STDERR to some other file.

## 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

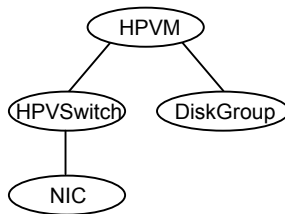
The HPVirtualMachine agent has the following limitations:

- 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

The agent functions for this agent follow:

- |         |   |
|---------|---|
| Online  | Uses the <code>hpvmstart</code> command to start the virtual machine (VMGuest).                                   |
| Offline | Attempts a graceful shut down of the virtual machine. Uses the <code>hpvmstop</code> command to stop the VMGuest. |

Monitor	<p>Uses the <code>hvvmstatus</code> command to detect the virtual machine's state. Returns the following status:</p> <ul style="list-style-type: none"> <li>■ If the Virtual Machine is missing, it returns an UNKNOWN state.</li> <li>■ If the Virtual Machine is running, then it returns an ONLINE state.</li> <li>■ If the Virtual Machine is not running, it returns OFFLINE.</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	<p>The virtual machine (VMGuest) name that the agent monitors. This attribute is unique.</p> <p>Type-dimension: string-scalar</p> <p>Example: "vmg_01"</p>

**Table 5-12** Required attributes (*continued*)

Required attribute	Description
DelayAfterOnline	<p>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.</p> <p>Type-dimension: integer-scalar</p> <p>Default: 30</p>

**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
```

## Service group with 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 = dgl  
    )  
    HPVSwitch pub0 (  
        VSwitchName = pub0  
    )  
    HPVirtualMachine hpvm (  
        VMName = vcsivml  
    )  
    NIC nic (  
        Device = lan0  
    )  
    Volume vol (  
        Volume = voll  
        DiskGroup = dgl  
    )  
    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
//     }
//   }
// }
```

## Debug log levels

The HPVirtualMachine agent uses the following debug log levels:  
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.

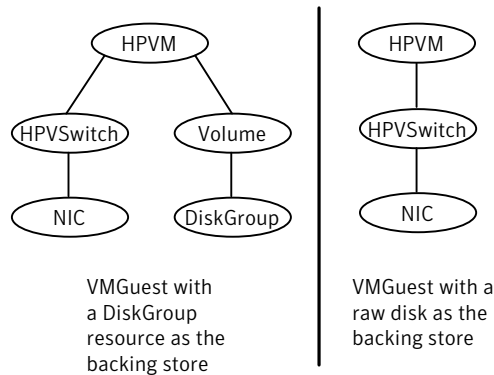
---

For agent limitations and requirements, See [“HPVSwitch agent notes”](#) on page 195.

## 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 HPVVirtualMachine resource.

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



## Agent functions

The agent functions for this agent follow:

Online	Starts the virtual switch using the native <code>hpvmnet</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>hpvmnet</code> command. After the shutdown of VMGuest, no network is required. The native <code>hpvmnet</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 ONLINE.</li><li>■ If the virtual switch is not running or is halted, it returns OFFLINE.</li><li>■ If the virtual switch is not present on the system, it returns the state UNKNOWN.</li></ul>
Clean	Stops the virtual switch. It is similar to the offline operation.

## State definitions

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

## Attributes

**Table 5-14** Required attributes

Required attribute	Description
VSwitchName	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](#)
- [Requirements](#)

### Agent limitation

This agent does not manage the virtual NIC on VMGuest.

### Requirements

For the HPVSwitch agent 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:

- See [“Creating an Integrity virtual machine service group”](#) on page 196.
- See [“Service group with VxVM volume as the backing store”](#) on page 197.
- See [“Service group with raw disk as the backing store”](#) on page 198.

### 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 HPVirtualMachine 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 HPVirtualMachine 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 to use 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 HPVirtualMachine resource *vm* and the child Volume resource *vol*.
- Ensure that all resources are enabled before bringing them online.

## Service group with 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 = vcsivm1
    )
    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
```

## Debug log levels

The HPVSwitch agent uses the following debug log levels:

DBG\_3, DBG\_4

# Infrastructure and support agents

This chapter includes the following topics:

- [About the infrastructure and support agents](#)
- [NotifierMngr agent](#)
- [Proxy agent](#)
- [Phantom agent](#)
- [RemoteGroup agent](#)

## 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.

Refer to the *Admin 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 for HP-UX

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 SntpServer is not specified; otherwise, SnmpConsoles is an optional attribute. Specify both SnmpConsoles and SntpServer if desired.</p> <p>Type and dimension: string-association</p> <p>Example:</p> <p>"172.29.10.89" = Error, "172.29.10.56" = Information</p>

**Table 6-1** Required attributes for HP-UX (*continued*)

Required attribute	Description
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                      Example: "smtp.example.com"</p>

**Table 6-2** Optional attributes for HP-UX

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                      Default: 14141</p>
MessagesQueue	<p>Size of the VCS engine's message queue. Minimum value is 30.</p> <p>Type and dimension: integer-scalar                      Default: 30</p>
NotifierListeningPort	<p>Any valid, unused TCP/IP port number.</p> <p>Type and dimension: integer-scalar                      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                      Example: "10.209.77.111"</p>
SmtFromPath	<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                      Example: "usera@example.com"</p>

**Table 6-2** Optional attributes for HP-UX (*continued*)

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:</p> <p>"james@example.com" = SevereError, "admin@example.com" = Warning</p>
Smtprreturnpath	<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 VRFY, then set the Smtprvrfyoff to 1 in order for the Smtprreturnpath 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>
Smtprservervrfyoff	<p>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 Smtprserver attribute while sending emails.</p> <p>Type and dimension: boolean-scalar</p> <p>Default: 0</p>
Snmprcommunity	<p>Specifies the community ID for the SNMP manager.</p> <p>Type and dimension: string-scalar</p> <p>Default: public</p>

**Table 6-2** Optional attributes for HP-UX (*continued*)

Optional attribute	Description
SnmpdTrapPort	Port on the SNMP console machine where SNMP traps are sent.  If you specify more than one SNMP console, all consoles use this value.  Type and dimension: integer-scalar  Default: 162

## Resource type definition

```
type NotifierMngr (  
    static int RestartLimit = 3  
    static str ArgList[] = { EngineListeningPort, MessagesQueue,  
        NotifierListeningPort, NotifierSourceIP, SnmpdTrapPort,  
        SnmpCommunity, SnmpConsoles, SmtperServer, SmtperServerVrfyOff,  
        SmtperServerTimeout, SmtperReturnPath, SmtperFromPath, SmtperRecipients  
    }  
  
    int EngineListeningPort = 14141  
    int MessagesQueue = 30  
    int NotifierListeningPort = 14144  
    int SnmpdTrapPort = 162  
    str SnmpCommunity = public  
    str SnmpConsoles{}  
    str SmtperServer  
    boolean SmtperServerVrfyOff = 0  
    int SmtperServerTimeout = 10  
    str SmtperReturnPath  
    str SmtperReturnPath  
    str SmtperFromPath  
    str SmtperRecipients{}  
)
```

## 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

Configuration for HP-UX follows:

```
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"  
    SmtRecipients = { "vcsadmin@your_company.com" =  
        SevereError }  
)
```

```
ntfr requires nicproxy
```

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

## Proxy agent

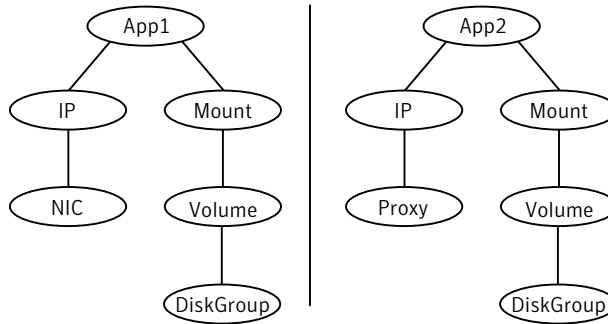
The Proxy agent mirrors the state of another resource on a local or remote system. It provides a means to specify and modify one resource and have its state reflected by its proxies. You can use the agent when you need to replicate the status of a resource.

A Proxy resource can only point to None or OnOnly type of resources, and can reside either in a failover or a parallel group. A target resource and its proxy cannot be in the same group.

## Dependencies

No dependencies exist for the Proxy resource.

**Figure 6-1** Sample service group that includes a Proxy resource



## Agent functions

**Monitor** Determines status based on the target resource status.

## Attributes

**Table 6-3** Required attribute

Required attribute	Description
TargetResName	Name of the target resource that the Proxy resource mirrors. The target resource must be in a different resource group than the Proxy resource. Type and dimension: string-scalar Example: "nic1"

**Table 6-4** Optional attribute

Optional attribute	Description
TargetSysName	Mirrors the status of the TargetResName attribute on systems that the TargetSysName variable specifies. If this attribute is not specified, the Proxy resource assumes the system is local. Type and dimension: string-scalar Example: "sysa"

## Resource type definition

```
type Proxy (
    static str ArgList[] = { TargetResName, TargetSysName,
        "TargetResName:Probed", "TargetResName:State" }
    static int OfflineMonitorInterval = 60
    static str Operations = None
    str TargetResName
    str TargetSysName
)
```

## Sample configurations

### Configuration 1

```
Proxy proxy1 (
    TargetResName = "nic1"
)
```

### Configuration 2

The proxy resource mirrors the state of the resource nic2 on sysa.

```
Proxy proxy1(
    TargetResName = "nic2"
    TargetSysName = "sysa"
)
```

### Configuration 3

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 192.98.16.255"  
RouteOptions@sysa = "default 192.98.16.103 0"  
RouteOptions@sysb = "default 192.98.16.104 0"  
)  
IPMultiNIC ip1 (  
  Address = "192.98.16.78"  
  NetMask = "255.255.255.0"  
  MultiNICResName = mnic  
  Options = "broadcast 192.98.16.255"  
)
```

ip1 requires mnic

```
group grp2 (  
  SystemList = { sysa, sysb }  
  AutoStartList = { sysa }  
)  
IPMultiNIC ip2 (  
  Address = "192.98.16.79"  
  NetMask = "255.255.255.0"  
MultiNICResName = mnic  
  Options = "mtu 1500"  
)  
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 agent enables VCS to determine the status of parallel service groups that do not include OnOff resources, which are resources that VCS can start and stop. Without the "dummy" resource provided by this agent, VCS cannot assess the status of groups that only contain None (Persistent) and OnOnly resources because

the state of these resources is not considered in the process of determining whether a group is online. Refer to the *VCS Administrator's Guide* for information on categories of service groups and resources.

Do not use the Phantom resource in failover service groups.

Also, the Phantom resource should not be used in service groups that don't contain any resources.

---

**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.

## Attribute

The attribute for the following agent:

## 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 refer to 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. For more information: See <a href="#">Table 6-5</a> on page 213.
Offline	Takes the remote service group offline. For more information: See <a href="#">Table 6-5</a> on page 213.

Monitor	<p>Monitors the state of the remote service group.</p> <p>The true state of the remote service group is monitored only on the online node in the local cluster. For more information:</p> <p>See <a href="#">Table 6-5</a> on page 213.</p>
Clean	<p>If the RemoteGroup resource faults, the Clean function takes the remote service group offline. For more information:</p> <p>See <a href="#">Table 6-5</a> on page 213.</p>

## State definitions

ONLINE	<p>Indicates that the remote service group is in an ONLINE state.</p> <p>If the ReturnIntOffline attribute is not set to RemotePartial, then the remote service group is either in an ONLINE or PARTIAL state.</p>
OFFLINE	<p>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.</p> <p>The RemoteGroup resource returns intentional offline if the attribute ReturnIntOffline is set to an appropriate value.</p>
FAULTED	<p>Indicates that the RemoteGroup resource has unexpectedly gone offline.</p>
UNKNOWN	<p>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.</p>

## 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>

**Table 6-5** Required attributes (*continued*)

Required attribute	Description
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="#">Table 6-6</a> on page 217.</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: "DBGrp"</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 (*continued*)

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> <p>■ OnOff</p> <p>The RemoteGroup resource brings the remote service group online or takes it offline.</p> <p>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.</p> </li> <li> <p>■ MonitorOnly</p> <p>The RemoteGroup resource only monitors the state of the remote service group. The RemoteGroup resource cannot online or offline the remote service group.</p> <p>Make sure that you bring the remote service group online before you online the RemoteGroup resource.</p> </li> <li> <p>■ OnlineOnly</p> <p>The RemoteGroup resource only brings the remote service group online. The RemoteGroup resource cannot take the remote service group offline.</p> <p>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.</p> </li> </ul> <p>Type and dimension: string-scalar</p>

**Table 6-5** Required attributes (*continued*)

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 (*continued*)

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. 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>
ToleranceLimit MonitorInterval	<p>If you expect the RemoteGroup agent to tolerate sudden offlines of the remote service group, then modify the ToleranceLimit attribute.</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 (*continued*)

Type level attributes	Description
ExternalStateChange	<p>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.</p> <p>See the <i>Veritas Cluster Server Administrator's Guide</i> for more information about these attributes.</p>

## 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 includes the following topics:

- [About the testing agents](#)
- [ElifNone agent](#)
- [FileNone agent](#)
- [FileOnOff agent](#)
- [FileOnOnly agent](#)

## 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

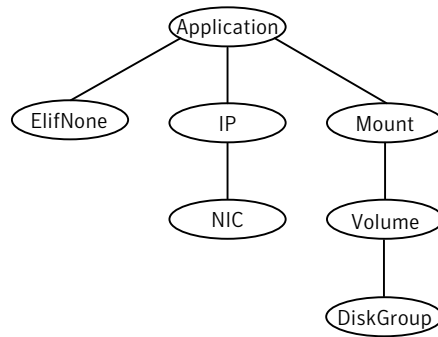
The ElifNone agent monitors a file. It checks for the file's absence.

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 for ElifNone agent

No dependencies exist for the ElifNone resource.

**Figure 7-1** Sample service group that includes an ElifNone resource



## Agent function for ElifNone agent

**Monitor** Checks for the specified file. If it exists, the resource faults. If it does not exist, the agent reports the resource as ONLINE.

## State definitions for ElifNone agent

**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 for ElifNone agent

**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 for ElifNone agent

```
type ElifNone (
    static str ArgList[] = { PathName }
    static int OfflineMonitorInterval = 60
    static str Operations = None
    str PathName
)
```

## Sample configuration for ElifNone agent

```
ElifNone tmp_file01 (
    PathName = "/tmp/file01"
)
```

## Debug log levels for ElifNone agent

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.

See ["Dependencies for FileNone agent"](#) on page 222.

See ["Agent functions for FileNone agent"](#) on page 223.

See ["State definitions for FileNone agent"](#) on page 223.

See ["Attributes for FileNone agent"](#) on page 224.

See ["Resource type definition for FileNone agent"](#) on page 224.

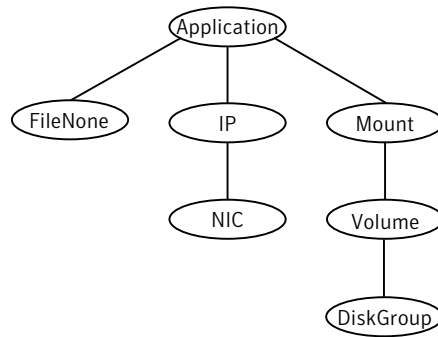
See ["Sample configuration for FileNone agent"](#) on page 224.

See ["Debug log levels for FileNone agent"](#) on page 224.

## Dependencies for FileNone agent

No dependencies exist for the FileNone resource.

**Figure 7-2** Sample service group that includes an FileNone resource



See [“FileNone agent”](#) on page 222.

## Agent functions for FileNone agent

Monitor	Checks for the specified file. If it exists, the agent reports the resource as ONLINE. If it does not exist, the resource faults.
---------	---

See [“FileNone agent”](#) on page 222.

## State definitions for FileNone agent

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.

See [“FileNone agent”](#) on page 222.

## Attributes for FileNone agent

**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"

See [“FileNone agent”](#) on page 222.

## Resource type definition for FileNone agent

```
type FileNone (  
    static str ArgList[] = { PathName }  
    static int OfflineMonitorInterval = 60  
    static str Operations = None  
    str PathName  
)
```

See [“FileNone agent”](#) on page 222.

## Sample configuration for FileNone agent

```
FileNone tmp_file01 (  
    PathName = "/tmp/file01"  
)
```

See [“FileNone agent”](#) on page 222.

## Debug log levels for FileNone agent

The FileNone agent uses the following debug log levels:

DBG\_1, DBG\_4, DBG\_5

See [“FileNone agent”](#) on page 222.

## FileOnOff agent

The FileOnOff agent creates, removes, and monitors a file.

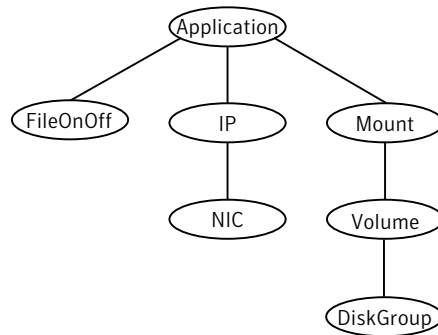


You can use the FileNone agent to test service group behavior. You can also use it as an "impostor" resource, where it takes the place of a resource for testing.

## Dependencies for FileOnOff agent

No dependencies exist for the FileOnOff resource.

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



## Agent functions for FileOnOff agent

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	Removes the specified file forcibly when necessary.

## State definitions for FileOnOff agent

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 for FileOnOff agent

**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 for FileOnOff agent

```
type FileOnOff (  
    static str ArgList[] = { PathName }  
    str PathName  
)
```

## Sample configuration for FileOnOff agent

```
FileOnOff tmp_file01 (  
    PathName = "/tmp/file01"  
)
```

## Debug log levels for FileOnOff agent

The FileOnOff agent uses the following debug log levels:

DBG\_1, DBG\_4, DBG\_5

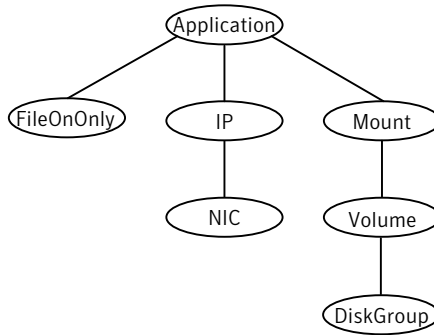
## FileOnOnly agent

The FileOnOnly agent creates and monitors a file.

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 for FileOnOnly agent

No dependencies exist for the FileOnOnly resource.

**Figure 7-4** Sample service group that includes a FileOnOnly resource

## Agent functions for FileOnOnly agent

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 for FileOnOnly agent

The state definitions for this agent follow:

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 for FileOnOnly agent

**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 for FileOnOnly agent

```
type FileOnOnly (  
    static str ArgList[] = { PathName }  
    static str Operations = OnOnly  
    str PathName  
)
```

## Sample configuration for FileOnOnly agent

```
FileOnOnly tmp_file02 (  
    PathName = "/tmp/file02"  
)
```

## Debug log levels for FileOnOnly agent

The FileOnOnly agent uses the following debug log levels:

DBG\_1, DBG\_4, DBG\_5

# Replication agents

This chapter includes the following topics:

- [About the replication agents](#)
- [RVG agent](#)
- [RVGPrimary agent](#)
- [RVGSnapshot](#)
- [RVGShared agent](#)
- [RVGLogowner agent](#)
- [RVGSharedPri agent](#)

## About the replication agents

Use the replication agents to provide high availability for VVR resources.

Refer to the *Veritas Storage Foundation and High Availability Solutions Replication Administrator's Guide* for information on configuring the Replication agents for high availability.

## RVG agent

Brings the RVG online, monitors read and write access to the RVG, and takes the RVG offline. This is a failover resource. The RVG agent enables replication between clusters. It manages the Primary VVR node in one cluster and the Secondary VVR node in another cluster. Each node can be failed over in its respective cluster. In this way, replication is made highly available.

The RVG agent manages the state of the RVG during local failovers. The RVGPrimary agent manages the role of the RVG during a wide area failover.

Using a VCS global cluster enables you to fail over the Primary role from a Primary VVR node to a Secondary VVR node.

The RVG agent includes the following key features:

- Removes potential single points of failure by enabling Primary and Secondary VVR nodes to be clustered.
- Enables you to bring a service group online to start VCS-managed applications that use VVR.
- Continues replication after a node in a cluster fails without losing updates.
- Ensures that VVR can be added to any VCS cluster by including the RVG resource type definitions.

An example configuration file for this agent that can be used as a guide when creating your configuration is located at:

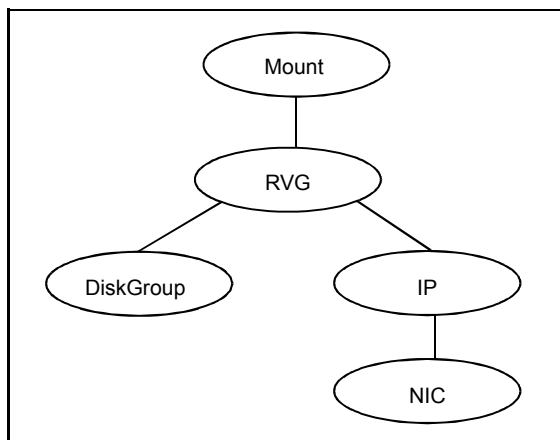
```
/etc/VRTSvcs/conf/sample_vvr/RVG
```

## Dependencies for RVG agent

The RVG resource represents the RVG (Replicated Volume Group) in the RDS (Replicated Data Set). The RVG resource is dependent on the DiskGroup resource. The RVG resource is also dependent on the IP resources that it uses for replication.

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

**Figure 8-1** Sample service group for an RVG resource



## Agent functions for RVG agent

The RVG agent has the following agent functions:

Online	Verifies whether the DiskGroup agent has recovered the RVG. If not, recovers and starts the data volumes and the Storage Replicator Log (SRL), recovers the RVG, recovers all RLINKs in the RVG, and then starts the RVG.
Offline	Stops the RVG.
Monitor	Monitors the state of the RVG using the <code>vxprint</code> command.  The RVG resource monitors an RVG for local access only. It does not monitor replication.
Clean	Stops the RVG.
Info	The info entry point displays information about the replication status of a RDS.

## State definitions for RVG agent

The RVG agent has the following state definitions:

ONLINE	Indicates that the RVG is in <code>ENABLED/ACTIVE</code> state.
OFFLINE	Indicates that the RVG is in <code>DISABLED/CLEAN</code> state.
FAULTED	The RVG resource fails if the RVG is not in the <code>ENABLED/ACTIVE</code> state.

## Attributes for RVG agent

**Table 8-1** Required attributes

Required attributes	Description
RVG	The name of the RVG being monitored. Type and dimension: string-scalar Example: "hr_rvg"
DiskGroup	The disk group that this RVG is associated with. Type and dimension: string-scalar Example: "hrbg"

**Table 8-1** Required attributes (*continued*)

Required attributes	Description
StorageDG	The name of the bunker disk group. Type and dimension: string-scalar Example: "hr_bdg"
StorageRVG	The name of the bunker RVG. Type and dimension: string-scalar Example: "hr_brvg"
StorageHostIds	A space-separated list of the hostids of each node in the bunker cluster. Type and dimension: string-keylist Example: "bunker_host"

## Resource type definitions for RVG agent

The RVG agent resource type definition follows.

```
type RVG (  
    static int NumThreads = 1  
    static str ArgList[] = { RVG, DiskGroup }  
    str RVG  
    str DiskGroup  
    str StorageRVG  
    str StorageDG  
    str StorageHostIds  
)
```

## Sample configurations for RVG agent

```
RVG rvg (  
    RVG = ApplicationRVG  
    DiskGroup = vvrldg  
    StorageRVG = ApplicationRVG  
    StorageDG = vvrldg  
    StorageHostIds = "bunker_host"  
)
```



## RVGPrimary agent

The RVGPrimary agent enables migration and takeover of a VVR Replicated Volume Group (RVG) in a VCS environment. Bringing a resource of type RVGPrimary online causes the RVG on the local host to become a primary.

The agent is useful when hosts in both the primary and secondary side are clustered, in particular a VCS replicated data cluster or a VCS global cluster, to completely automate the availability of writable replicated disks to a VCS-managed application.

The RVGPrimary agent includes the following features:

- Removes the manual steps of migrating a VVR primary and secondary roles when failing over applications across a wide area.
- Minimizes the need for resynchronizing replicated volumes by attempting a migration before attempting a hard takeover.
- Waits for the two sides of a replicated data set to become completely synchronized before migrating roles.
- Supports an automatic fast failback resynchronization of a downed primary if it later returns after a takeover.
- Allows you to distinguish the Primary site after network failure or disaster
- After a successful migration or takeover of a Secondary RVG, the RVGPrimary agent ensures to automatically start the replication from the new Primary to any additional Secondary(s) that exists in the RDS.
- Before a takeover, the RVGPrimary agent synchronizes the Secondary site with any bunker associated with the Primary site, when the Primary site is not available.

Refer to the *Veritas Storage Foundation and High Availability Solutions Replication Administrator's Guide* for information on configuring the Replication agents for high availability.

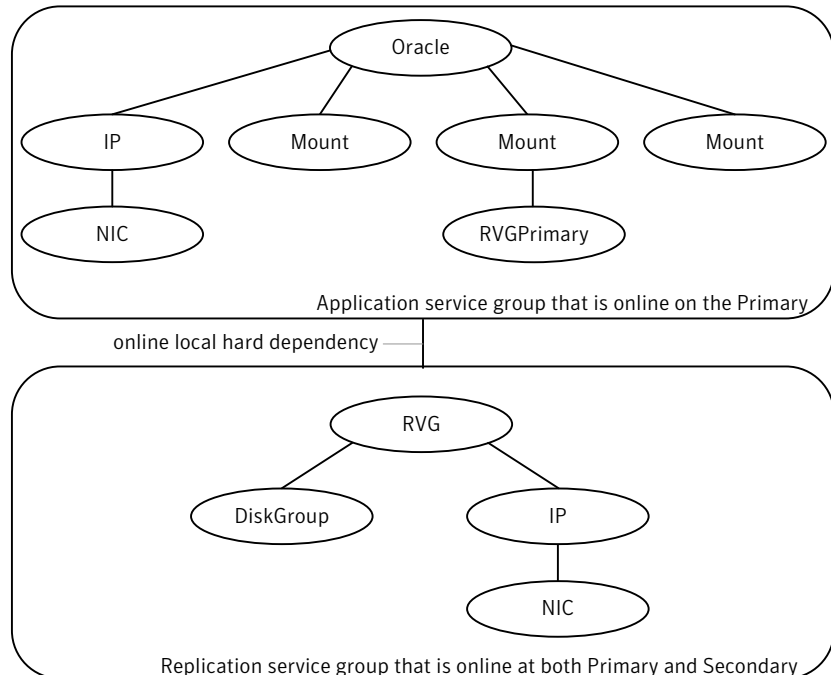
A sample configuration file for this agent that you can use as a guide to create the configuration is located at `/etc/VRTSvcs/conf/sample_vvr/RVGPrimary`.

## Dependencies for RVGPrimary agent

You usually use the RVGPrimary agent in conjunction with the RVG agent in two groups with an online local hard group dependency. The parent group contains the resources that manage the actual application and file systems and as the RVGPrimary resource. The child group contains the resources managing the storage infrastructure, which include the RVG and DiskGroup type resources.

Refer to the *Veritas Storage Foundation and High Availability Solutions Replication Administrator's Guide* for information about the setup of a VVR environment using the RVGPrimary agent.

**Figure 8-2** Sample service group for an RVGPrimary resource



## Agent functions for RVGPrimary agent

The RVGPrimary agent has the following agent functions:

- Online**      Determines the current role of the RVG. If the role is Secondary it attempts a migration. It waits for any outstanding writes from the original Primary. If the original Primary is down, it attempts a takeover. You can configure the RVGPrimary agent so that, before a takeover, the agent synchronizes the Secondary site with any bunker associated with the Primary site, when the Primary site is not available. If the RVG is a Primary, it performs no actions and goes online.
- Offline**      Performs no actions.
- Monitor**      Performs no actions. The RVG agents monitors the actual RVG.
- Clean**        Performs no actions.

fbsync	<p>This is an action entry point.</p> <p>It resynchronizes the original Primary with the new Primary that has taken over with fast-failback, after the original Primary had become unavailable. This needs to be executed when the original Primary becomes available and starts acting as a Secondary.</p>
ElectPrimary	<p>This is an action entry point.</p> <p>It can be executed to retain the specified RVG as the Primary in a Primary-Primary configuration.</p> <p>For more details, refer to the <i>Veritas Storage Foundation and High Availability Solutions Replication Administrator's Guide</i>.</p>

## State definitions for RVGPrimary agent

The RVGPrimary agent has the following state definitions:

ONLLINE	Indicates that the role of the RVG is Primary.
FAULTED	The RVG agents monitors the actual RVG. Accidental migration of a VVR Primary outside of VCS causes other resources to fault immediately, such as Mount. No special monitoring by this agent is necessary.

## Attributes for RVGPrimary agent

**Table 8-2** Required attributes

Required attributes	Description
RvgResourceName	<p>The name of the RVG resource type that this agent promotes. The name RVG resource type which has been configured using the RVG agent.</p> <p>Type and dimension: string-scalar</p>
AutoTakeover	<p>A flag to indicate whether the agent should perform a takeover on online if the original Primary is down.</p> <p>AutoTakeover and AutoResync are mutually exclusive attributes.</p> <p>Type and dimension: integer-scalar</p>

**Table 8-2** Required attributes (*continued*)

Required attributes	Description
AutoResync	<p>Indicates whether the agent should attempt to automatically perform a fast-failback resynchronization of the original Primary after a takeover and after the original Primary returns.</p> <p>You can use the following values for this attribute:</p> <ul style="list-style-type: none"> <li>0—instructs the agent to not attempt to perform a fast-failback resynchronization of the original Primary after a takeover and after the original Primary returns.</li> <li>1—instructs the agent to attempt to automatically perform a fast-failback resynchronization of the original Primary after a takeover and after the original Primary returns.</li> </ul> <p>AutoTakeover and AutoResync are mutually exclusive attributes.</p> <p>Type and dimension: integer-scalar</p>

## Resource type definitions for RVGPrimary agent

The RVGPrimary resource type definition follows.

```

type RVGPrimary (
    static keylist SupportedActions = { fbsync, electprimary }
    static int NumThreads = 1
    static int OnlineRetryLimit = 1
    static str ArgList[] = { RvgResourceName, "RvgResourceName:RVG",
        "RvgResourceName:DiskGroup", AutoTakeover, AutoResync,
        BunkerSyncTimeOut, BunkerSyncElapsedTime }
    str RvgResourceName
    int AutoTakeover = 1
    int AutoResync = 0
    str BunkerSyncTimeOut
    int BunkerSyncElapsedTime = 0
)

```

## Sample configurations for RVGPrimary agent

```

RVGPrimary rvg-pri (
    RvgResourceName = rvgRes
)

```

# RVGSnapshot

For a fire drill, creates and destroys a transactionally consistent space-optimized snapshot of all volumes in a VVR secondary replicated data set. The RVGSnapshot agent takes space-optimized snapshots on a secondary RVG. These snapshots can be mounted and written to without affecting the actual replicated data, which means that the space-optimized snapshot can be an effective tool for scheduling a “fire drill” to confirm that a wide-area failover is possible. By combining this agent with the VCS Mount agent, the CFMount agent, and VCS agents that manage the application being replicated, you can create a special fire drill service group. You can bring this service group online and take it offline at regularly scheduled intervals to verify that the disaster recovery environment is robust.

In addition to the agent itself, a text-based wizard `/opt/VRTSvcs/bin/fdsetup` that prepares the VVR and VCS infrastructure for a fire drill and a script `/opt/VRTSvcs/bin/fdsched` that runs the fire drill and consolidates the results are also included.

Complete details are in the *Veritas Cluster Server Administrator's Guide*.

The RVGSnapshot agent includes the following key features:

- Automates the process of creating a space-optimized snapshot on a VVR secondary that can be mounted to simulate a wide-area failover without affecting the production application.
- Includes a wizard to effectively set up and schedule fire drills that are completely managed by VCS.

---

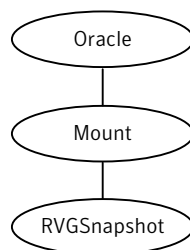
**Note:** The RVGSnapshot agent does not support Volume Sets.

---

## Dependencies for RVGSnapshot

The RVGSnapshot agent depends on these resources.

**Figure 8-3** Sample service group for an RVGSnapshot resource



## Agent functions for RVGSnapshot

The RVGSnapshot agent has the following agent functions:

Online	Creates a transactionally consistent snapshot of all volumes in the RVG.
Offline	Destroys the snapshot.
Monitor	No operation; failure of the snapshot will be indicated by the failure of the Mount resource of any file systems mounted on it.
Clean	Cleans up any failed snapshot creation or deletion.

## State definitions for RVGSnapshot

The RVGSnapshot agent has the following state definitions:

ONLINE	Indicates that a snapshot was created.
OFFLINE	Indicates that a snapshot was destroyed.
FAULTED	The RVGSnapshot resource faults on timeout if a snapshot creation did not succeed during an online.

## Attributes for RVGSnapshot

**Table 8-3** Required attributes

Required attributes	Description
RvgResourceName	The name of the VCS RVG-type resource that manages the RVG that will be snapshot by this agent. Type and dimension: string-scalar
CacheObj	Name of the cache object that is required for a space-optimized snapshot; the fdsetup wizard will create one if it does not exist Type and dimension: string-scalar
Prefix	Token put before the name of the actual volume when creating the snapshotted volumes. Type and dimension: string-scalar

**Table 8-4** Optional attributes

Optional attributes	Description
DestroyOnOffline	<p>A flag to indicate whether to destroy the snapshot upon taking the resources offline. For a fire drill, the snapshot should be deleted to reduce any performance impact of leaving the snapshot for a long period of time; however, if there is interest in keeping the data, then this value should be set to 0. The default is 1 (true).</p> <p>Type and dimension: integer-scalar</p> <p>Default: 1</p>
FDFile	<p>The fire drill schedule updates this attribute with the system name and the path to a file containing the output of the last complete fire drill for the group containing an RVGSnapshot resource.</p> <p>Type and dimension: string-scalar</p>

## Resource type definitions for RVGSnapshot

The resource type definition for the RVGSnapshot agent follows.

```

type RVGSnapshot (
    static keylist RegList = { Prefix }
    static int NumThreads = 1
    static str ArgList[] = { RvgResourceName, CacheObj, Prefix,
        DestroyOnOffline }
    str RvgResourceName
    str CacheObj
    str Prefix
    boolean DestroyOnOffline = 1
    temp str FDFile
)

```

## Sample configurations for RVGSnapshot

```

RVGSnapshot rvg-sos (
    RvgResourceName = ApplicationRVG
    CacheObj = cacheobj
    Prefix = snap
)

```

## RVGShared agent

Monitors the RVG in a shared environment. This is a parallel resource. The RVGShared agent enables you to configure parallel applications to use an RVG in a cluster. The RVGShared agent monitors the RVG in a shared disk group environment. The RVGShared agent must be configured as a parallel group in VCS. Typically, the RVGShared resource is online or offline at the same time on all the nodes in the VCS cluster. An example configuration file for this agent that can be used as a guide when creating your configuration is located at `/etc/VRTSvcs/conf/sample_vvr/RVGLogowner`.

## Dependencies for RVGShared agent

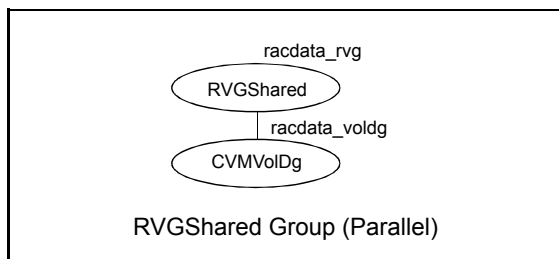
The RVGShared resource represents the RVG of the RDS. The RVGShared resource is dependent on the CVMVolDg resource.

The RVGShared resource must be configured in a parallel group.

Refer to the *Veritas Storage Foundation and High Availability Solutions Replication Administrator's Guide* for information on configuring parallel applications for highly availability.

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

**Figure 8-4** Sample service group for an RVGShared resource



---

**Note:** Do not add any volumes that are part of the RVG in the CVMVolume attribute of the CVMVolDg resource. The volumes in the RVG are managed by the RVGShared resource.

---

## Agent functions for RVGShared agent

The RVGShared agent has the following agent functions:



Online	Verifies whether the RVG is started. If the RVG is not started, recovers and starts the RVG.
Offline	No action.
Monitor	Displays the state as <code>ONLINE</code> if the RVG is started. Displays the state as <code>OFFLINE</code> if the RVG is not started.
Clean	No action.
Info	The info entry point displays information about the replication status of a RDS.

## State definitions for RVGShared agent

The RVGShared agent has the following state definitions:

ONLINE	Indicates that the RVG is in the <code>ENABLED/ACTIVE</code> state.
OFFLINE	Indicates that the RVG is not in the <code>ENABLED/ACTIVE</code> state or that the administrator has invoked the offline entry point.

## Attributes for RVGShared agent

**Table 8-5** Required attributes

Required attributes	Description
RVG	The name of the RVG being monitored. Type and dimension: string-scalar
DiskGroup	The shared-disk group with which this RVG is associated. Type and dimension: string-scalar

## Resource type definitions for RVGShared agent

The RVGShared resource type definition follows.

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

## Sample configurations for RVGShared agent

```
RVGShared racdata_rvg (  
RVG = racl_rvg  
DiskGroup = oradatadg  
)
```

## RVGLogowner agent

Assigns and unassigns a node as the logowner in the CVM cluster; this is a failover resource. The RVGLogowner agent assigns or unassigns a node as a logowner in the cluster. To replicate data, VVR requires network connectivity between the Primary and the Secondary. In a shared disk group environment, only one node, that is, the logowner, can replicate data to the Secondary.

For replication to be highly available, the logowner must be highly available. To make the logowner highly available, the RVGLogowner resource must be configured as a resource in a failover group. Also, a virtual IP must be set up on the logowner to enable replication and failover of the logowner from one node to another in a cluster. The virtual IP must be configured as an IP resource.

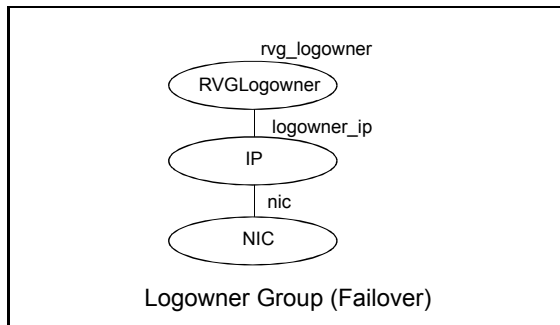
For more information about the logowner, see the *Veritas Storage Foundation and High Availability Solutions Replication Administrator's Guide*. An example configuration file for this agent that can be used as a guide when creating your configuration, is located at `/etc/VRTSvcs/conf/sample_vvr/RVGLogowner`.

## Dependencies for RVGLogowner agent

The RVGLogowner resource represents the logowner for RVG in the cluster. The RVGLogowner resource is dependent on the IP resource that it uses for replication.

The RVGLogowner resource must be configured in a failover group. The RVGLogowner group is used in conjunction with the RVGSharedPri and RVGShared agents in separate groups, with the appropriate service group dependencies.

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

**Figure 8-5** Sample service group for an RVGLogowner resource

## Agent functions RVGLogowner agent

The RVGLogowner agent has the following agent functions:

Online	Assigns the logowner on the node.
Offline	Unassigns the logowner on the node.
Monitor	Returns <code>ONLINE</code> if the node is the logowner and the RVG is in <code>ENABLED/ACTIVE</code> state. Returns <code>OFFLINE</code> if the node is the logowner and the state is not <code>ENABLED/ACTIVE</code> , or if the node is not the logowner (regardless of the state). The RVG for which the logowner is monitored must be configured as the <code>RVGShared</code> resource type.
Clean	Unassigns the logowner on the node.

## State definitions RVGLogowner agent

The RVGLogowner agent has the following state definitions:

ONLINE	Indicates that the node is the logowner for the RVG in the cluster.
OFFLINE	Indicates that the node is not the logowner for the RVG in the cluster.

## Attributes RVGLogowner agent

**Table 8-6** Required attributes

Required attributes	Description
RVG	The name of the RVG being monitored. Type and dimension: string-scalar Example: "hr_rvg"
DiskGroup	The disk group with which this RVG is associated. Type and dimension: string-scalar Example: "hrbg"

**Table 8-7** Internal attributes

Bunker attributes	Description
StorageDG	For internal use only, do not modify. The name of the bunker disk group. Type and dimension: string-scalar Example: "hr_bdg"
StorageRVG	For internal use only, do not modify. The name of the bunker RVG. Type and dimension: string-scalar Example: "hr_brvg"
StorageHostIds	For internal use only, do not modify. A space-separated list of the hostids of each node in the bunker cluster. Type and dimension: string-keylist Example: "bunker_host"

## Resource type definitions RVGLogowner agent

The RVGLogowner resource type definition follows.

```
type RVGLogowner (
    static int NumThreads = 1
    static str ArgList[] = { RVG, DiskGroup }
    static int OnlineRetryLimit = 5
    str RVG
    str DiskGroup
```

```
str StorageRVG
str StorageDG
str StorageHostIds
)
```

## RVGLogowner agent notes RVGLogowner agent

The RVGLogowner agent has the following notes:

### **CVM master node needs to assume the logowner role for VCS managed VVR resources**

If you use VCS to manage RVGLogowner resources in an SFCFSHA environment or an SF Oracle RAC environment, Symantec recommends that you perform the following procedures. These procedures ensure that the CVM master node always assumes the logowner role. Not performing these procedures can result in unexpected issues that are due to a CVM slave node that assumes the logowner role.

For a service group that contains an RVGLogowner resource, change the value of its PreOnline trigger to 1 to enable it.

#### **To enable the PreOnline trigger from the command line on a service group that has an RVGLogowner resource**

- ◆ On each node in the cluster, perform the following command:

```
# hagrps -modify RVGLogowner_resource_sg PreOnline 1 -sys system
```

Where *RVGLogowner\_resource\_sg* is the service group that contains the RVGLogowner resource. The *system* is the name of the node where you want to enable the trigger.

On each node in the cluster, merge the preonline\_vvr trigger into the default triggers directory.

### To merge the preonline\_vvr trigger

- ◆ On each node in the cluster, merge the preonline\_vvr trigger to the /opt/VRTSvcs/bin/triggers directory.

```
# cp /opt/VRTSvcs/bin/sample_triggers/VRTSvcs/preonline_vvr \  
/opt/VRTSvcs/bin/triggers
```

Refer to the sample configurations directory for samples of how to enable these triggers (/opt/VRTSvcs/bin/sample\_triggers/VRTSvcs.)

---

**Note:** In case you perform a master switch using the `vxclustadm setmaster` command, the RVGLogowner service group will continue to remain online on the original master, even when the PreOnline attribute is enabled.

Run the following command in order to switch the RVGLogowner service group to the new master node:

```
# hagr -switch RVGLogowner_resource_sg -to new_cvm_master
```

Where `RVGLogowner_resource_sg` is the name of the service group that contains the RVGLogowner resource and `new_cvm_master` is the name of the new CVM master node.

---

## Sample configurations RVGLogowner agent

```
RVGLogowner vvr_rvglogowner (  
  RVG = app_rvg  
  DiskGroup = vvrdbg  
)
```

## RVGSharedPri agent

Attempts to migrate or takeover a Secondary to a Primary when a parallel service group fails over. The RVGSharedPri agent enables migration and takeover of a VVR replicated data set in parallel groups in a VCS environment. Bringing a resource of type RVGSharedPri online causes the RVG on the local host to become a primary if it is not already. The agent is useful when hosts in both the primary and secondary side are clustered using a VCS global cluster, to completely automate the availability of writable replicated disks to an application managed by VCS.

You cannot use the primary-elect feature with this agent. For a detailed description of the primary-elect feature, see *Veritas Storage Foundation and High Availability Solutions Replication Administrator's Guide*.

The RVGSharedPri agent includes the following key features:

- Removes manual steps of migrating a VVR primary and secondary roles when failing over applications across a wide area.
- Minimizes the need for resynchronizing replicated volumes by attempting a migration before attempting a hard takeover.
- Waits for the two sides of a replicated data set to become completely synchronized before migrating roles.
- Supports an automatic fast fallback resynchronization of a downed primary if it later returns after a takeover.

Sample configuration files are located in the `/etc/VRTSvcs/conf/sample_rac/` directory and include CVR in the filename. These sample files are installed as part of the VRTSdbac depot, and can be used as a guide when creating your configuration.

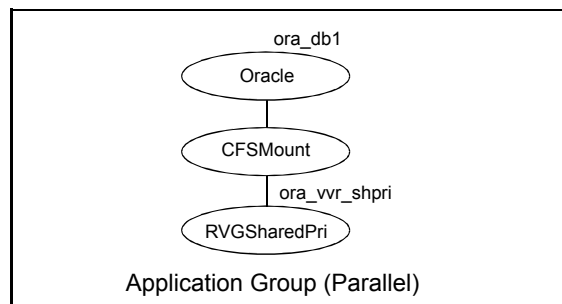
## Dependencies for RVGSharedPri agent

The RVGSharedPri agent is used in conjunction with the RVGShared and RVGLogowner agents in separate groups, with the appropriate service group dependencies.

Refer to the *Veritas Storage Foundation and High Availability Solutions Replication Administrator's Guide* for information on configuring parallel applications for highly availability.

The RVGSharedPri agent must be configured in a parallel service group. The application service group contains the resources managing the actual application and file systems as well as the RVGSharedPri agent.

**Figure 8-6** Sample service group for an RVGSharedPri resource



## Agent functions for RVGSharedPri agent

The RVGSharedPri agent has the following agent functions:

Online	Determines the current role of the RVG; if Secondary, attempt a migrate, waiting for any outstanding writes from the original Primary; if the original Primary is down attempt a takeover; if the RVG is a Primary, perform no actions and go online
Offline	Performs no actions.
Monitor	Performs no actions; monitoring of the actual RVG is done by the RVGShared agent.
Clean	Performs no actions.
fbsync	This is an action entry point.  It resynchronizes the original Primary with the new Primary that has taken over with fast-failback, after the original Primary had become unavailable.  This needs to be executed when the original Primary becomes available and starts acting as a Secondary.
resync	This is an action entry point.  It resynchronizes the Secondaries with the Primary using DCM.

## State definitions for RVGSharedPri agent

The RVGSharedPri agent has the following state definitions:

FAULTED	Monitoring of the actual RVG is done by the RVGShared agent; accidental migration of a VVR Primary outside of VCS would cause other resources to fault immediately, such as Mount, so no special monitoring by this agent is necessary.
---------	---

## Attributes for RVGSharedPri agent

**Table 8-8** Required attributes

Required attributes	Description
RvgResourceName	The name of the RVGShared resource type that this agent will promote, that is, the name RVG resource type which has been configured using the RVGShared agent.  Type and dimension: string-scalar



**Table 8-8** Required attributes (*continued*)

Required attributes	Description
AutoTakeover	A flag to indicate whether the agent should perform a takeover on online if the original Primary is down.  Type and dimension: integer-scalar  Default: 1
AutoResync	A flag to indicate whether the agent should attempt to automatically perform a fast-failback resynchronization of the original Primary after a takeover and after the original Primary returns.  Type and dimension: integer-scalar  Default: 0
VCSResLock	This attribute is reserved for internal use by VCS.  Type and dimension: string-scalar

## Resource type definitions for RVGSharedPri agent

The RVGSharedPri resource type definition follows.

```
type RVGSharedPri (
    static keylist SupportedActions = { fbsync, resync }
    static int NumThreads = 1
    static int OnlineRetryLimit = 1
    static str ArgList[] = { RvgResourceName, "RvgResourceName:RVG",
        "RvgResourceName:DiskGroup", AutoTakeover, AutoResync }
    str RvgResourceName
    int AutoTakeover = 1
    int AutoResync = 0
    temp str VCSResLock
)
```

## Sample configurations for RVGSharedPri agent

```
RVGSharedPri ora_vvr_shpri (
    RvgResourceName = racdata_rvg
    OnlineRetryLimit = 0
)
```

# Index

## Symbols

802.1Q trunking 78

## A

about

Network agents 77

Samba agents 144

agent

modifying 18

agent functions 142

Apache HTTP server agent 157

Application agent 168

CoordPoint agent 178

DiskGroup agent 23

DiskGroupSnap agent 33

DNS agent 116

ElifNone agent 221

FileNone agent 223

FileOnOff agent 225

FileOnOnly agent 227

HPVirtualMachine agent 189

HPVSwitch agent 194

IP agent 80

IPMultiNIC agent 90

IPMultiNICB agent 102

LVMCombo agent 59

LVMLogicalVolume agent 53

LVMVolumeGroup agent 56

Mount agent 62

MultiNICA agent 94

MultiNICB agent 109

NetBIOS agent 152

NFS agent 133

HP-UX 133

NFSRestart agent 136

NIC agent 86

NotifierMngr agent 201

Phantom agent 210

Process agent 182

ProcessOnOnly agent 186

Proxy agent 207

agent functions (*continued*)

RemoteGroup agent 212

SambaServer agent 146

SambaShare agent 150

Share agent

HP-UX 142

Volume agent 47

VolumeSet agent 50

agents. *See* RVG agent

Apache HTTP server 156

Application 166

CoordPoint 177

DiskGroup 22

DiskGroupSnap 31

DNS 115

ElifNone 220

FileNone 222

FileOnOff 224

FileOnOnly 226

HPVirtualMachine 189

HPVSwitch 193

IP 79

IPMultiNIC 89

IPMultiNICB 101

LVMCombo 58

LVMLogicalVolume 52

LVMVolumeGroup 55

Mount 62

MultiNICA 93

NetBIOS 152

NFS 132

NFSRestart 135

NIC 84

NotifierMngr 200

Phantom 209

Process 181

ProcessOnOnly 186

Proxy 206

RemoteGroup 211

RVGLogowner. *See* RVGLogowner agent

RVGPrimary. *See* RVGPrimary agent

RVGShared. *See* RVGShared agent

**agents** *(continued)*

- RVGSharedPri. See RVGSharedPri agent
- RVGSnapshot. See RVGSnapshot agent
- SambaServer 146
- SambaShare 149
- Share 141
- Volume 47

**Apache HTTP server agent**

- agent functions 157
- attributes 159
- description 156
- detecting application failure 164
- sample configuration 165
- state definitions 158

**Application agent**

- agent functions 168
- description 166
- high availability fire drill 167
- resource type definition 174
- sample configurations 176
- state definitions 170

**association dimension 18****attribute data types 18****attributes**

- CoordPoint agent 179
- DiskGroup agent 25
  - HP-UX 25
- DiskGroupSnap agent 34
- DNS agent 118
- ElifNone agent 221
- FileNone agent 224
- FileOnOff agent 226
- FileOnOnly agent 228
- HPVirtualMachine agent 190
- HPVSwitch agent 195
- IP agent 80
  - HP-UX 80
- IPMultiNIC agent 91
  - HP-UX 91
- IPMultiNICB agent 103
  - HP-UX 103
- LVMCombo agent 60
- LVMLogicalVolume agent 53
  - HP-UX 53
- LVMVolumeGroup agent 57
  - HP-UX 57
- modifying 18
- Mount agent
  - HP-UX 65

**attributes** *(continued)*

- MultiNICA agent 94
  - MultiNICB agent 109
  - NFS agent 134
    - HP-UX 134
  - NFSRestart agent 138
    - HP-UX 138
  - NIC agent 86
    - HP-UX 86
  - NotifierMngr agent 201
    - HP-UX 201
  - Phantom agent 210
  - Process agent 183
    - HP-UX 183
  - Proxy agent 207
  - RemoteGroup agent 213
  - SambaServer agent 147
  - Share agent 143
    - HP-UX 143
  - Volume agent 48
  - VolumeSet agent 51
- AutoResync attribute**
- RVGPrimary agent 249

**B**

- boolean data types 18

**C**

- Checklist to ensure the proper operation of
  - MultiNICB 101
- Cluster Manager (Java Console)
  - modifying attributes 18
- CNAME record 127
- configuration files
  - main.cf 211
  - modifying 18
  - types.cf 18
- CoordPoint agent
  - agent functions 178
  - attributes 179
  - description 177
  - resource type definition 179
  - sample configurations 180
  - state definitions 178

**D**

- data type
  - boolean 18

- data type *(continued)*
  - string 18
- data types
  - integer 18
- dependency graphs
  - RVGLogowner agent 242
  - RVGPrimary agent 233
  - RVGShared agent 240
  - RVGSharedPri agent 247
- description
  - resources 18
- dimensions
  - keylist 18
  - scalar 18
  - vector 18
- DiskGroup agent
  - agent functions 23
  - attributes 25
  - description 22
  - high availability fire drill 29
  - HP-UX attributes 25
  - resource type definition 28
  - sample configurations 30
  - state definitions 25
- DiskGroupSnap agent
  - agent functions 33
  - attributes 34
  - description 31
  - resource type definition 39
  - sample configurations 39
  - state definitions 33
- DNS agent 117
  - agent functions 116
  - attributes 118
  - description 115
  - resource type definition 124
  - sample web server configuration 127

## E

- ElifNone agent
  - agent functions 221
  - attributes 221
  - description 220
  - resource type definition 222
  - sample configuration 222
  - state definitions 221

## F

- failover group
  - RVGLogowner agent 242
- fast failback
  - AutoResync attribute of RVGPrimary 249
- fast failback resynchronization
  - RVGPrimary 233
  - RVGSharedPri 247
- FileNone agent
  - agent functions 223
  - attribute 224
  - description 222
  - resource type definition 224
  - sample configurations 224
  - state definitions 223
- FileOnOff agent
  - agent functions 225
  - attribute 226
  - description 224
  - state definitions 225
- FileOnOnly agent
  - agent functions 227
  - attribute 228
  - description 226
  - resource type definition 228
  - sample configuration 228
  - state definitions 227
- fire drill
  - RVGSnapshot agent 237

## H

- high availability fire drill 29, 72, 79, 84, 127, 139, 167, 181
- HP Auto Port Aggregation (APA) support 85, 108
- HP-UX attributes 103
- HPVirtualMachine agent
  - agent functions 189
  - attributes 190
  - description 189
  - resource type definition 191
  - sample configurations 191
  - state definitions 190
- HPVSwitch agent
  - agent functions 194
  - attributes 195
  - description 193
  - resource type definition 195
  - sample configurations 196
  - state definitions 195

**I**

- integer data types 18
- IP agent
  - agent functions 80
  - attributes 80
  - description 79
  - high availability fire drill 79
  - HP-UX attributes 80
  - resource type definitions 82
  - sample configurations 83
  - state definitions 80
- IPMultiNIC agent
  - agent functions 90
  - attributes 91
  - description 89
  - HP-UX attributes 91
  - resource type definitions 92
  - state definitions 90
- IPMultiNICB agent 103, 106
  - agent functions 102
  - attributes 103
  - description 101
  - manually migrating IP address 106
  - requirements 102
  - resource type definition 105
  - state definitions 103

**K**

- keylist dimension 18

**L**

- logowner
  - virtual IP requirement 242
- LVMCombo agent
  - agent functions 59
  - attributes 60
  - description 58
  - resource type definition 60
  - sample configurations 61
  - state definitions 59
- LVMLogicalVolume agent
  - agent functions 53
  - attributes 53
  - description 52
  - HP-UX attributes 53
  - resource type definition 54
  - sample configurations 55
  - state definitions 53

- LVMVolumeGroup agent
  - agent functions 56
  - attributes 57
  - description 55
  - HP-UX attributes 57
  - resource type definition 57
  - state definitions 56

**M**

- main.cf 18, 211
- main.xml 18
- migrating
  - RVGPrimary 233
  - RVGSharedPri 246
- modifying
  - configuration files 18
- modifying agents 18
- monitor scenarios
  - DNS agent 127
- Mount agent
  - agent functions 62, 64
  - description 62
  - high availability fire drill 72, 127, 139
  - HP-UX attributes 65
  - notes 72
  - resource type definition 71
  - sample configurations 75
- MultiNICA agent 98
  - agent functions 94
  - attributes 94
  - description 93
  - resource type attributes 97
  - RouteOptions 98
  - sample configurations 99
  - state definitions 94
- MultiNICB agent 109
  - agent functions 109
  - attributes 109
  - resource type definition 113
  - state definitions 109

**N**

- NetBIOS agent
  - agent functions 152
  - description 152
  - resource type definition 153
  - sample configurations 155
  - state definitions 153

NFS agent  
 agent functions 133  
   HP-UX 133  
 attributes 134  
   HP-UX 134  
 description 132  
 resource type definition 134  
 sample configurations 135  
 state definitions 133

NFSRestart agent  
 agent functions 136  
 attributes 138  
   HP-UX 138  
 description 135  
 resource type definition 138  
 sample configuration 140  
 state definitions 137

NIC agent  
 agent functions 86  
 attributes 86  
   HP-UX 86  
 description 84  
 high availability fire drill 84  
 resource type definitions 88  
 sample configurations 88  
 state definitions 86

noautoimport flag 30

NotifierMngr agent  
 agent functions 201  
 attributes 201  
 description 200  
 HP-UX attributes 201  
 resource type definition 204  
 sample configurations 204  
 state definitions 201

## O

online query 127

## P

parallel group  
   RVGShared agent 240

Phantom agent  
 agent functions 210  
 attributes 210  
 description 209  
 resource type definition 210  
 sample configurations 211

prerequisites  
   Samba agents 144

Process agent  
 agent functions 182  
 attributes 183  
 description 181  
 high availability fire drill 181  
 HP-UX attributes 183  
 resource type definition 184  
 sample configurations 185  
 state definitions 183

ProcessOnOnly agent  
 agent functions 186  
 description 186  
 resource type definition 188  
 state definitions 187

Proxy agent  
 agent functions 207  
 attributes 207  
 description 206  
 resource type definition 208  
 sample configurations 208

## R

RemoteGroup agent  
 agent functions 212  
 attributes 213  
 description 211  
 resource type definition 219  
 state definitions 213

resource type definition 49  
   SambaShare agent 151

resource type definitions  
 Application agent 174  
 CoordPoint agent 179  
 DiskGroup agent 28  
 DiskGroupSnap agent 39  
 DNS agent 124  
 ElifNone agent 222  
 FileNone agent 224  
 FileOnOnly agent 228  
 HPVirtualMachine agent 191  
 HPVSwitch agent 195  
 IP agent 82  
 IPMultiNIC agent 92  
 IPMultiNICB agent 105  
 LVMCombo agent 60  
 LVMLogicalVolume agent 54  
 LVMVolumeGroup agent 57

resource type definitions *(continued)*

- Mount agent 71
- MultiNICA agent 97
- MultiNICB agent 113
- NetBIOS agent 153
- NFS agent 134
- NFSRestart agent 138
- NIC agent 88
- NotifierMngr agent 204
- Phantom agent 210
- Process agent 184
- ProcessOnOnly agent 188
- Proxy agent 208
- RemoteGroup agent 219
- SambaServer agent 148
- Share agent 143
- Volume agent 49

## resources

- description of 18

## RVG agent

- described 229

## RVGLogowner agent

- dependency graph 242
- described 242
- failover group 242

## RVGPrimary agent

- dependency graph 233
- described 233
- migrating 233
- takeover 233

## RVGShared agent

- dependency graph 240
- described 240
- parallel group 240

## RVGSharedPri agent

- dependency graph 247
- described 246
- migrating 246
- takeover 246

## RVGSnapshot agent

- described 237
- fire drill 237

**S**

## Samba agents 144

- overview 144
- prerequisites 144

## SambaServer agent

- agent functions 146

SambaServer agent *(continued)*

- attributes 147
- description 146
- resource type definition 148
- sample configuration 149
- state definitions 146

## SambaShare agent 149

- agent functions 150
- attributes 150
- resource type definition 151
- sample configurations 151
- state definitions 150

## sample configurations 106

- Apache HTTP server agent 165

- Application agent 176

- CoordPoint agent 180

- DiskGroup agent 30

- DiskGroupSnap agent 39

- ElifNone agent 222

- FileNone agent 224

- FileOnOff agent 226

- FileOnOnly agent 228

- HPVirtualMachine agent 191

- HPVSwitch agent 196

- IP agent 83

- IPMultiNICB agent 106

- LVMCombo agent 61

- LVMLogicalVolume agent 55

- Mount agent 75

- MultiNICA agent 99

- NetBIOS agent 155

- NFS agent 135

- NFSRestart agent 140

- NIC agent 88

- NotifierMngr agent 204

- Phantom agent 211

- Process agent 185

- Proxy agent 208

- SambaServer agent 149

- SambaShare agent 151

- Share agent 144

- Volume agent 49

## scalar dimension 18

- secure DNS update 128

## Share agent 142

- agent functions 142

- HP-UX 142

- attributes 143

- description 141

- Share agent (*continued*)
  - HP-UX attributes 143
  - resource type definitions 143
  - sample configurations 144
  - state definitions 142
- snapshots
  - using RVGSnapshot agent for 237
- State definitions
  - VolumeSet agent 50
- state definitions 117
  - Apache HTTP server agent 158
  - Application agent 170
  - CoordPoint agent 178
  - DiskGroup agent 25
  - DiskGroupSnap agent 33
  - DNS agent 117
  - ElifNone agent 221
  - FileNone agent 223
  - FileOnOff agent 225
  - FileOnOnly agent 227
  - HPVirtualMachine agent 190
  - HPVSwitch agent 195
  - IP agent 80
  - IPMultiNIC agent 90
  - IPMultiNICB agent 103
  - LVMCombo agent 59
  - LVMLogicalVolume agent 53
  - LVMVolumeGroup agent 56
  - Mount agent 64
  - MultiNICA agent 94
  - MultiNICB agent 109
  - NetBIOS agent 153
  - NFS agent 133
  - NFSRestart agent 137
  - NIC agent 86
  - NotifierMngr agent 201
  - Process agent 183
  - ProcessOnOnly agent 187
  - RemoteGroup agent 213
  - SambaServer agent 146
  - SambaShare agent 150
  - Share agent 142
  - Volume agent 48
- string data type 18

## T

- takeover
  - RVGPrimary 233
  - RVGSharedPri 246

- trigger script 114
- trunking 78
- types.cf 18

## V

- VCS
  - resource types 18
- vector dimension 18
- virtual IP
  - RVGLogowner agent requirement 242
- Volume agent
  - agent functions 47
  - attributes 48
  - description 47
  - sample configurations 49
  - state definitions 48
- volume sets 29
- VolumeSet agent
  - agent functions 50
  - attributes 51
  - State definitions 50