Veritas™ Cluster Server Bundled Agents Reference Guide

Solaris

5.1 Service Pack 1

Symantec

Veritas Cluster Server Bundled Agents Reference Guide

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

Product version: 5.1 SP1

Document version: 5.1.SP1.2

Legal Notice

Copyright © 2011 Symantec Corporation. All rights reserved.

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

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

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

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

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

Technical Support

Symantec Technical Support maintains support centers globally. Technical Support's primary role is to respond to specific queries about product features and functionality. The Technical Support group also creates content for our online Knowledge Base. The Technical Support group works collaboratively with the other functional areas within Symantec to answer your questions in a timely fashion. For example, the Technical Support group works with Product Engineering and Symantec Security Response to provide alerting services and virus definition updates.

Symantec's support offerings include the following:

- A range of support options that give you the flexibility to select the right amount of service for any size organization
- Telephone and/or web-based support that provides rapid response and up-to-the-minute information
- Upgrade assurance that delivers automatic software upgrades protection
- Global support purchased on a regional business hours or 24 hours a day, 7 days a week basis
- Premium service offerings that include Account Management Services

For information about Symantec's support offerings, you can visit our web site at the following URL:

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

All support services will be delivered in accordance with your support agreement

and the then-current enterprise technical support policy.

Contacting Technical Support

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

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

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

When you contact Technical Support, please have the following information

available:

- Product release level
- Hardware information
- Available memory, disk space, and NIC information
- Operating system
- Version and patch level

- Network topology
- Router, gateway, and IP address information
- Problem description:
  - Error messages and log files
  - Troubleshooting that was performed before contacting Symantec
  - Recent software configuration changes and network changes

Licensing and registration

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

www.symantec.com/business/support/

Customer service

Customer service information is available at the following URL:

www.symantec.com/business/support/

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

- Questions regarding product licensing or serialization
- Product registration updates, such as address or name changes
- General product information (features, language availability, local dealers)
- Latest information about product updates and upgrades
- Information about upgrade assurance and support contracts
- Information about the Symantec Buying Programs
- Advice about Symantec's technical support options
- Nontechnical presales questions
- Issues that are related to CD-ROMs or manuals

Support agreement resources

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

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

Documentation

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

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

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

docs@symantec.com

About Symantec Connect

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

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

Contents

Chapter 1 Introduction

Resources and their attributes 19
Modifying agents and their resources 20
Attributes 20
Zone-aware agents 21
Enabling debug log messages 22

Chapter 2 Storage agents

About the storage agents 23
DiskGroup agent 24
- Dependencies 24
- Agent functions 24
- State definitions 26
- Attributes 27
- Resource type definition 31
- DiskGroup agent notes 31
  - High availability fire drill 31
  - Using volume sets 32
  - Setting the noautoimport flag for a disk group 32
  - Configuring the Fiber Channel adapter 33
- Sample configurations 33
  - DiskGroup resource configuration 33
- Debug log levels 33

DiskGroupSnap agent 34
- Dependencies 34
- Agent functions 35
- State definitions 35
- Attributes 36
- DiskGroupSnap agent notes 36
  - Configuring the SystemZones attribute for the fire drill service group 37
  - Configuring the firedrill service group 37
  - Adding the ReuseMntPt attribute to the ArgList attribute for the Mount agent type 37
  - Configuration considerations 38

Agent limitations 39
Resource type definition 39
Sample configurations 40
Typical main.cf configuration 41
Oracle main.cf configuration 43
Debug log levels 46
Disk agent 47
Agent functions 47
State definitions 47
Attributes 48
Resource type definition 48
Debug log levels 48
Volume agent 49
Dependencies 49
Agent functions 49
State definitions 49
Attributes 50
Resource type definition 51
Sample configuration 51
Debug log levels 51
VolumeSet agent 52
Dependencies 52
Agent functions 52
State definitions 52
Attributes 53
Resource type definition 53
Sample configurations 54
A configured VolumeSet that is dependent on a DiskGroup resource 54
Agent notes 54
Inaccessible volumes prevent the VolumeSet agent from coming online 54
Debug log levels 54
Mount agent 55
Dependencies 55
Agent functions 56
State definitions 58
Attributes 59
Resource type definition 66
Mount agent notes 67
High availability fire drill 67
VxFS file system lock 67
IMF usage notes 68

IPv6 usage notes 68
Support for loopback file system 69
ZFS file system and pool creation example 70
Enabling second level monitoring for the Mount agent 70
Sample configurations 71
VxFS configuration example 71
Debug log levels 71
Zpool agent 72
Limitations 72
Dependencies 72
Agent functions 73
State definitions 74
Attributes 75
Resource type definition 76
Sample configurations 76
Debug log levels 77

Chapter 3 Network agents

About the network agents 79
Agent comparisons 79
IP and NIC agents 79
IPMultiNIC and MultiNICA agents 80
IPMultiNICB and MultiNICB agents 80
802.1Q trunking 81
Link aggregation support 81
IP agent 82
High availability fire drill 82
Dependencies 82
Agent functions 83
State definitions 84
Attributes 85
Resource type definition 87
Sample configurations 88
Configuration 1 88
NetMask in decimal (base 10) 88
Configuration of NetMask in hexadecimal (base 16) 88
Debug log levels 88
NIC agent 89
High availability fire drill 89
Dependencies 89
Agent functions 90
State definitions 90
Attributes 91

Resource type definition 92
Sample configurations 93
Configuration without network hosts (using default ping mechanism) 93
Configuration with network hosts 93
IPv6 configuration 93
Exclusive IP Zone configuration 94
Debug log levels 95
IPMultiNIC agent 96
Dependencies 96
Agent functions 97
State definitions 97
Attributes 98
Resource type definition 99
Sample configuration: IPMultiNIC and MultiNICA 100
Debug log levels 101
MultiNICA agent 102
Dependencies 102
Agent function 103
State definitions 103
Attributes 104
Resource type definition 107
MultiNICA notes 108
Using RouteOptions 108
Sample configurations 109
MultiNICA and IPMultiNIC 109
IPv6 configuration 110
Debug log levels 110
About the IPMultiNICB and MultiNICB agents 111
Checklist to ensure the proper operation of MultiNICB 111
IPMultiNICB agent 112
Dependencies 112
Requirements for IPMultiNICB 113
Agent functions 113
State definitions 114
Attributes 115
Resource type definition 117
Manually migrating a logical IP address 118
Sample configurations 118
Other sample configurations for IPMultiNICB and MultiNICB 118
Debug log levels 118
MultiNICB agent 119
Base and Multi-Pathing modes 119

Oracle trunking 119
The haping utility 119
Dependencies 120
Agent functions 120
State definitions 121
Attributes 121
Optional attributes for Base and Mpathd modes 122
Optional attributes for Base mode 123
Optional attributes for Multi-Pathing mode 126
Resource type definition 128
Solaris operating modes: Base and Multi-Pathing 128
Base mode 128
Multi-Pathing mode 130
Trigger script 130
Sample configurations 131
Interface configuration for Solaris 131
Setting up test IP addresses for Base Mode 131
IPMultiNICB and MultiNICB configuration 132
IPv6 configuration 133
Debug log levels 133
DNS agent 134
Dependencies 134
Agent functions 135
State definitions 136
Attributes 137
Resource type definition 141
DNS agent notes 141
High availability fire drill 142
Monitor scenarios 142
Sample Web server configuration 142
Secure DNS update for BIND 9 143
Setting up secure updates using TSIG keys for BIND 9 143
Sample configurations 144
Basic IPv6 configuration 144
IPv6 CNAME sample configuration 145
IPv4 A sample configuration 145
Debug log levels 145

Chapter 4 File share agents

About the file service agents 147
NFS agent 148
Dependencies 148
Agent functions 149

State definitions 149
Attributes 149
Resource type definition 151
NFS agent notes 151
Using Service Management Facility (SMF) to control NFS daemons for Solaris 10 151
Sample configurations 152
Debug log levels 152
NFSRestart agent 153
Dependencies 153
Agent functions 154
State definitions 155
Attributes 156
Resource type definition 157
NFSRestart agent notes 157
About high availability fire drill 157
Mounting NFS export with the -vers options can cause lock failure 158
Providing a fully qualified host name 158
Service Management Facility—Solaris 10 159
Sample configurations 160
Basic agent configurations 160
Debug log levels 161
Share agent 162
Dependencies 162
Agent functions 162
State definitions 163
Attributes 164
Resource type definition 164
Share agent notes 164
High availability fire drill 164
Sample configurations 165
Debug log levels 165
About the Samba agents 166
The Samba agents 166
Before using the Samba agents 166
Supported versions 167
Notes for configuring the Samba agents 167
Configuring multiple SambaServer resources 167
Configuring Samba for non-standard configuration files or non-standard lock directories 167
SambaServer agent 168
Dependencies 168

Agent functions 168
State definitions 169
Attributes 170
Resource type definitions 172
Sample configurations 172
Debug log levels 172
SambaShare agent 173
Dependencies 173
Agent functions 173
State definitions 173
Attributes 174
Resource type definition 174
Sample configuration 175
Debug log levels 175
NetBios agent 176
Dependencies 176
Agent functions 176
State definitions 177
Attributes 177
Resource type definition 179
Sample configuration 179
Debug log levels 180

Chapter 5 Service and application agents

About the service and application agents 181
Apache Web server agent 182
Dependencies 182
Agent functions 183
State definitions 183
Attributes 184
Resource type definition 188
Apache Web server notes 189
Tasks to perform before you use the Apache Web server agent 189
About detecting application failure 190
About bringing an Apache Web server online outside of VCS control 190
About high Availability fire drill 190
Sample configurations 191
Basic configuration for Solaris 191
Basic IPv6 configuration 193
Application agent 194
High availability fire drill 194
Dependencies 195

Agent functions 196
State definitions 197
Attributes 198
Resource type definition 201
Application agent notes 202
Using Application agent with IMF 202
Sample configurations 202
Configuration 1 202
Configuration 2 203
Configuration 3 for Solaris 10 203
Debug log levels 203
CoordPoint agent 204
Dependencies 204
Agent functions 204
State definitions 205
Attributes 205
Resource type definition 206
Notes for the CoordPoint agent 206
CoordPoint agent I/O fencing reporting activities 206
AutoStartList attribute 206
Sample configuration 207
Debug log levels 207
Process agent 208
High availability fire drill 208
Dependencies 208
Agent functions 209
State definitions 210
Attributes 211
Resource type definition 211
Sample configurations 212
Configuration 1 212
Configuration 2 212
Debug log levels 212
ProcessOnOnly agent 213
Dependencies 213
Agent functions 213
State definitions 213
Attributes 214
Resource type definition 215
Sample configurations 215
Debug log levels 215
Zone agent 216
Dependencies 216

Agent functions 217
Attributes 217
Resource type definition 219
Sample configurations 219
Application resource in a non-global zone for Solaris 10 219
Debug log levels 220
LDom agent 221
Configuring primary and guest domain dependencies and failure policy 221
Dependencies 221
Network resources 221
Storage resources 222
Agent functions 222
State definitions 222
Attributes 223
Resource type definition 224
Sample configuration 224
Debug log levels 225
Project agent 226
Dependencies 226
Agent functions 226
Attributes 227
Resource type definition 227
Sample configuration 227
Debug log levels 228

Chapter 6 Infrastructure and support agents

About the infrastructure and support agents 229
NotifierMngr agent 230
Dependency 230
Agent functions 230
State definitions 230
Attributes 231
Resource type definition 234
Sample configuration 235
Configuration 235
Debug log levels 236
Proxy agent 237
Dependencies 237
Agent functions 237
Attributes 238
Resource type definition 239
Sample configurations 239

Configuration 1 239
Configuration 2 239
Configuration 3 239
Debug log levels 240
Phantom agent 241
Dependencies 241
Agent functions 241
Resource type definition 241
Sample configurations 241
Configuration 1 241
Configuration 2 242
RemoteGroup agent 243
Dependency 243
Agent functions 244
State definitions 244
Attributes 245
Resource type definition 250
Debug log levels 250

Chapter 7 Testing agents

About the testing agents 251
ElifNone agent 252
Dependencies 252
Agent function 252
State definitions 252
Attributes 253
Resource type definition 253
Sample configuration 253
Debug log levels 253
FileNone agent 254
Dependencies 254
Agent functions 254
State definitions 254
Attribute 255
Resource type definition 255
Sample configuration 255
Debug log levels 255
FileOnOff agent 256
Dependencies 256
Agent functions 256
State definitions 256
Attribute 257
Resource type definition 257

Sample configuration 257
Debug log levels 257
FileOnOnly agent 258
Dependencies 258
Agent functions 258
State definitions 258
Attribute 259
Resource type definition 259
Sample configuration 259
Debug log levels 259

Glossary 261

Index 263

Chapter 1

Introduction

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

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

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

Agents can:

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

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

Resources and their attributes

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

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

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

Modifying agents and their resources

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

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

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

Attributes

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

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

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

Table 1-1 Attribute data types

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

Table 1-1 Attribute data types

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

Table 1-2 Attribute dimensions

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

Zone-aware agents

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

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

Resource Type	RunInContainer	PassCInfo
Application	1	0
Apache	1	0
IP	0	1

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

Resource Type	RunInContainer	PassCInfo
IPMultiNIC	0	1
IPMultiNICB	0	1
Mount	0	0
NIC	0	1
Process	1	0
Zone	0	1

For more information on using zones in your VCS environment, refer to the *Veritas Storage Foundation and High Availability Solutions Virtualization Guide*.

Enabling debug log messages

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

To enable agent framework debug log messages:

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

For example:

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

To enable agent-specific debug log messages:

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

For example:

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

Alternatively, you can also use the following command:

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

Agent-specific debug log level information is specified in the agent's description. For example, for information about the Mount agent, see "Debug log levels" on page 86.

Chapter 2

Storage agents

This chapter contains:

- "About the storage agents" on page 23
- "DiskGroup agent" on page 24
- "DiskGroupSnap agent" on page 34
- "Disk agent" on page 47
- "Volume agent" on page 49
- "VolumeSet agent" on page 52
- "Mount agent" on page 55
- "Zpool agent" on page 72

About the storage agents

Use storage agents to Monitor shared storage.

DiskGroup agent

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

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

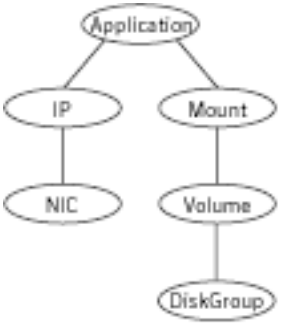
For important information on this agent, refer to:

"DiskGroup agent notes" on page 31

Dependencies

The DiskGroup resource does not depend on any other resources.

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



Agent functions

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

Clean

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

Info

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

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

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

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

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

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

Output includes:

```
DiskType sliced
FreeSize 35354136
```

Action

Different action agent functions follow:

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

See "High availability fire drill" on page 31.

State definitions

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

Attributes

Table 2-1 Required attributes

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

Table 2-2 Optional attributes

Optional attributes	Description
MonitorReservation	If the value is 1, and SCSI-3 fencing is used, the agent monitors the SCSI reservation on the disk group. If the reservation is missing, the monitor agent function takes the resource offline. Type and dimension: boolean-scalar Default: 0

Table 2-2 Optional attributes

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

Table 2-2 Optional attributes

Optional attributes	Description
StartVolumes	If value of this attribute is 1, the DiskGroup online function starts all volumes belonging to that disk group after importing the group. **Note:** With VxVM version 5.1.100.0 onwards, if the Veritas Volume Manager default autostartvolumes at system level is set to on, all the volumes of the disk group will be started as a part of the import disk group. Type and dimension: boolean-scalar Default: 1
StopVolumes	If value is 1, the DiskGroup offline function stops all volumes belonging to that disk group before it deports the disk group. Type and dimension: boolean-scalar Default: 1
UmountVolumes	This attribute enables the DiskGroup resource to forcefully go offline even if open volumes are mounted outside of VCS control. When the value of this attribute is 1 and the disk group has open volumes, the following occurs: ■ The agent attempts to unmount the file systems on open volumes. If required, the agent attempts to kill all VCS managed and un-managed applications using the file systems on those open volumes. ■ The agent attempts to forcefully unmount the file systems to close the volumes. Type and dimension: integer-scalar Default: 0

Table 2-2 Optional attributes

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

Table 2-3 Internal attributes

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

Resource type definition

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

DiskGroup agent notes

The DiskGroup agent has the following notes:

- "High availability fire drill" on page 31
- "Using volume sets" on page 32
- "Setting the noautoimport flag for a disk group" on page 32
- "Configuring the Fiber Channel adapter" on page 33

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.

VxVM versions 4.1 and 5.0 or later The Monitor function changes the value of the VxVM noautoimport flag from off to on. It changes the value instead of taking the service group offline. This action allows VCS to maintain control of importing the disk group.

The following command changes the autoimport flag to false:

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

For VxVM version 4.0

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

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

To deport the disk group, enter:

```
# vxdg deport disk_group
```

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

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

Configuring the Fiber Channel adapter

Most Fiber Channel (FC) drivers have a configurable parameter called “failover.” This configurable parameter is in the FC driver’s configuration file. This parameter is the number of seconds that the driver waits before it transitions a disk target from OFFLINE to FAILED. After the state becomes FAILED, the driver flushes all pending fiber channel commands back to the application with an error code. Symantec recommends that you use a non-zero value that is smaller than any of the MonitorTimeout values of the Disk Group resources. Use this value to avoid excessive waits for monitor timeouts.

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

Sample configurations

DiskGroup resource configuration

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

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

Debug log levels

The DiskGroup agent uses the following debug log levels:

DBG_1, DBG_4, DBG_5

DiskGroupSnap agent

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

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

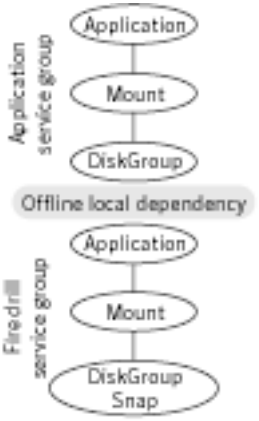
For important information about this agent, refer to:

"DiskGroupSnap agent notes" on page 36

Dependencies

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

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



Agent functions

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

State definitions

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

Attributes

Table 2-4 Required attributes

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

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

DiskGroupSnap agent notes

The DiskGroupSnap agent has the following notes:

■ "Configuring the SystemZones attribute for the fire drill service group" on page 37
■ "Configuring the firedrill service group" on page 37
■ "Adding the ReuseMntPt attribute to the ArgList attribute for the Mount agent type" on page 37
■ "Configuration considerations" on page 38
■ "Agent limitations" on page 39

Configuring the SystemZones attribute for the fire drill service group

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

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

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

Configuring the firedrill service group

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

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

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

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

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

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

1. Make the configuration read and write.

   ```
   # haconf -makerw
   ```

2. Add the ReuseMntPt attribute to the ArgList attribute.

   ```
   # hatype -modify Mount ArgList -add ReuseMntPt
   ```

3. Change the value of the ReuseMntPt attribute to 1 for the firedrill's Mount resource.

```
# hares -modify firedrill_mount_resource_name ReuseMntPt 1
```

4 Change the value of the ReuseMntPt attribute to 1 for the original Mount resource.

```
# hares -modify original_mount_resource_name ReuseMntPt 1
```

5 Make the configuration read only.

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

Configuration considerations

Keep the following recommendations in mind:

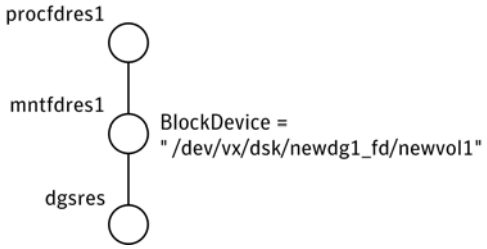
- You must install Veritas Volume Manager 5.1 or later with the FMR license and the Site Awareness license.
- Do not bring the DiskGroupSnap resource online in the SystemZone where the application service group is online.
- Make sure that the firedrill service group and the application service group both use the same values for the SystemZones attribute.
- Do not use Volume resources in the firedrill service group. The DiskGroupSnap agent internally uses the `vxvol` command to start all the volumes in the firedrill disk group.
- In large setups, you may need to tweak the various timer values so that the timers do not time out while waiting for VxVM commands to complete. The timers you need to tweak are the OfflineTimeout for the DiskGroupSnap resource and MonitorInterval and ActionTimeout for the associated DiskGroup resource, for example:

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

- When you create the firedrill service group, in general use the same attribute values that you use in the application service group.

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

Figure 2-3 Sample resource values for a DiskGroupSnap resource



Agent limitations

The following limitations apply to the DiskGroupSnap agent:

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

Resource type definition

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

Sample configurations

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

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

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

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

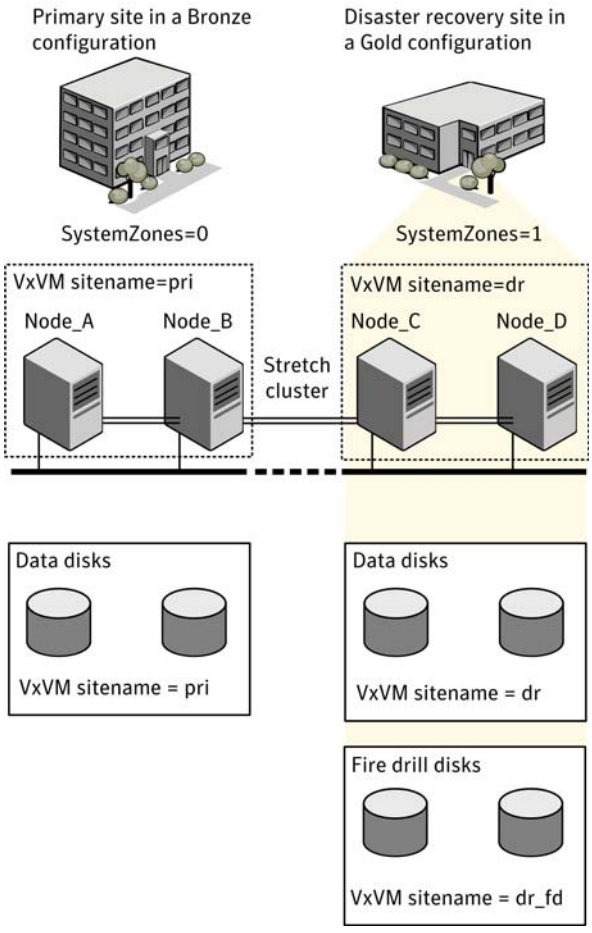
The Disaster Recovery site is in a Gold configuration as it has dedicated fire drill disks at the site. For the FDSiteName attribute, use the VxVM site tag given to the fire drill disks. For this configuration, the FDSiteName attribute values for the nodes at the Disaster recovery site follow:

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

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

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

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



Typical main.cf configuration

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

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

    DiskGroupSnap dgsres (
        TargetResName = dgres
```

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

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

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

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

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

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

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

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

    DiskGroup dgres (
        DiskGroup = newdg1
        )

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

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

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

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

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

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

Oracle main.cf configuration

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

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

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

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

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

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

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

        NIC fd_oranic (
                Device = NIC0
                )

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

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

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

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

        DiskGroup oradg_res (
                DiskGroup = oradg
                )

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

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

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

    NIC nic_Node_A4vip (
            Device = NIC0
            )

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

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

    Volume arch_vol (
            Volume = archive_vol
            DiskGroup = oradg
            )

    Volume data_vol (
            Volume = data_vol
            DiskGroup = oradg
            )

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

Debug log levels

The DiskGroupSnap agent uses the following debug log levels:

DBG_1

Disk agent

Monitors a physical disk or a partition.

You can use the Disk agent to monitor a physical disk or a slice that is exported to LDoms (available using LDoms 1.2 or later).

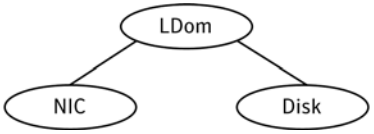
For LDoms with a physical disk or slice based boot image, a dependency must exist between the guest domain and primary domain. You configure the primary domain as the master of the guest domain. Perform the following:

- Set the failure-policy of primary (control) domain to stop. For example, in the primary domain enter the following command to set the dependent domain to stop when the primary domain faults:

  # **ldm set-domain failure-policy=stop primary**
- Set the primary domain as the master for the guest domain

  # **ldm set-domain master=primary *guestldom***

Figure 2-5 Sample service group that includes a Disk resource on Solaris



Agent functions

Monitor	Performs read I/O operations on the raw device to determine if a physical disk or a partition is accessible.

State definitions

ONLINE	Indicates that the disk is working normally.
FAULTED	Indicates that the disk has stopped working or is inaccessible.
UNKNOWN	Indicates that a problem exists either with the configuration or the ability to determine the status of the resource.

Attributes

Table 2-5 Required attributes

Required attribute	Description
Partition	Indicates which partition to monitor. Specify the partition with the full path beginning with a slash (/). For Solaris, if this path is not specified, the name is assumed to reside in /dev/rdsk/. Example: "/dev/rdsk/c2t0d0s2" Type and dimension: string-scalar

Resource type definition

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

Debug log levels

The Mount agent uses the following debug log levels:

DBG_1

Volume agent

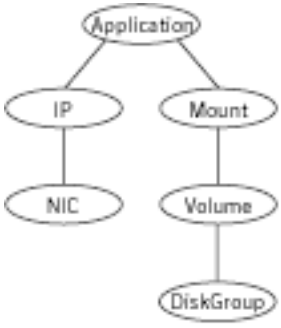
The Volume agent brings online, takes offline, and monitors a Veritas Volume Manager (VxVM) volume. Use the agent to make a volume highly available.

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

Dependencies

Volume resources depend on DiskGroup resources.

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



Agent functions

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

State definitions

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

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

Attributes

Table 2-6 Required attributes

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

Table 2-7 Internal attribute

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

Resource type definition

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

Sample configuration

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

Debug log levels

The Volume agent uses the following debug log levels:

DBG_1

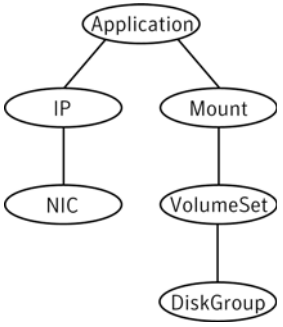
VolumeSet agent

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

Dependencies

VolumeSet resources depend on DiskGroup resources.

Figure 2-7 Sample service group that includes a VolumeSet resource



Agent functions

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

State definitions

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

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

Attributes

Table 2-8 Required attributes

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

Resource type definition

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

Sample configurations

This sections contains sample configurations for this agent.

A configured VolumeSet that is dependent on a DiskGroup resource

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

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

Agent notes

This sections contains notes about this agent.

Inaccessible volumes prevent the VolumeSet agent from coming online

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

To remove a volume from volume set

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

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

Debug log levels

The VolumeSet agent uses the following debug log levels:

DBG_1, DBG_4

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. The ContainerName and ContainerType attributes are deprecated.

This agent is zone-aware. The ContainerOpts resource type attribute for this type has a default value of 0 for RunInContainer and a default value of 0 for PassCInfo. Symantec recommends that you do not change these values. Refer to the *Storage Foundation High Availability Virtualization Guide* for more information.

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

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

The Mount agent supports the IPv6 protocol.

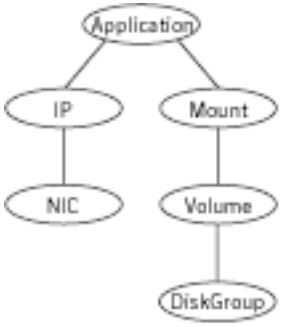
For important information about this agent, refer to:

"Mount agent notes" on page 67

Dependencies

The Mount resource does not depend on any other resources.

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



Agent functions

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

Info

The Mount info agent function executes the command:

```
df -k mount_point
```

The output displays Mount resource information:

```
Size Used Avail Use%
```

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

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

The command to retrieve information about the Mount resource is:

```
hares -value mountres ResourceInfo
```

Output includes:

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

Action

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

State definitions

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

Attributes

Table 2-9 Required attributes

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

Table 2-9 Required attributes

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

Table 2-9 Required attributes

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

Table 2-10 Optional attributes

Optional attribute	Description
MountOpt	Options for the `mount` command. Refer to the `mount` manual page for more information. Do not set the VxFS mount option "mntlock=*key*". The agent uses this option only when bringing a Mount resource online. Type and dimension: string-scalar Example: "rw"
SnapUmount	If the value of this attribute is 1, this attribute automatically unmounts VxFS snapshots when the file system is unmounted. If the value of this attribute is 0, and snapshots are mounted, the resource cannot be brought offline. In this case, failover does not occur. Type and dimension: integer-scalar Default: 0
CkptUmount	If the value of this attribute is 1, this attribute automatically unmounts VxFS Storage Checkpoints when file system is unmounted. If the value of this attribute is 0, and Storage Checkpoints are mounted, then failover does not occur. Type and dimension: integer-scalar Default: 1
SecondLevelMonitor	This attribute has been deprecated. Instead of this attribute, use the LevelTwoMonitorFreq attribute. For more information, see "Enabling second level monitoring for the Mount agent" on page 70.
SecondLevelTimeout	This attribute has been deprecated.

Table 2-10 Optional attributes

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

Table 2-10 Optional attributes

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

Table 2-10 Optional attributes

Optional attribute	Description
OptCheck	The value of this attribute determines if VCS should verify the mount options. The state of the resource is determined based on the result of the verification. If the value of this attribute is 0 (default), the mount options are not checked. If the value of the OptCheck attribute is 1, 2 or 3, a check is performed to see if the mount command options that you have specified for VCS are set in the MountOpt attribute. The MountOpt attributes should be the same as the actual mount command options. If the actual mount options differ from the MountOpt attribute, a message is logged. The state of the resource depends on the value of this attribute. If the value of the attribute is 1, the state of the resource is unaffected. If the value is 2, the state of the resource is set to offline. If the value is 3, state of the resource is set to unknown. Type and dimension: integer-scalar Default: 0
RecursiveMnt	If the value of this attribute is 1, VCS creates all the parent directories of the mount point if necessary. All directories in the path are created with system default permissions. **Note:** Permissions on mount points must be a minimum of 555 for the operating system commands to work correctly. Type and dimension: boolean-scalar Default: 0

Table 2-10 Optional attributes

Optional attribute	Description
ReuseMntPt	If the same mount point needs to be specified in more than one mount resource, set the value of this attribute to 1. Note that this attribute only accepts a value of 1 or 0. To use this attribute, the cluster administrator needs to add this attribute to the arglist of the agent. Set the appropriate group and resource dependencies such that only one resource can come online on a system at a time. Type and dimension: integer-scalar Default: 0

Resource type definition

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

Mount agent notes

The Mount agent has the following notes:

- "High availability fire drill" on page 67
- "VxFS file system lock" on page 67
- "IMF usage notes" on page 68
- "IPv6 usage notes" on page 68
- "Support for loopback file system" on page 69
- "ZFS file system and pool creation example" on page 70
- "Enabling second level monitoring for the Mount agent" on page 70

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/vfstab file

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

VxFS file system lock

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

```
# mount
```

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

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

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

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

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

IMF usage notes

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

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

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

IPv6 usage notes

Review the following information for IPv6 use:

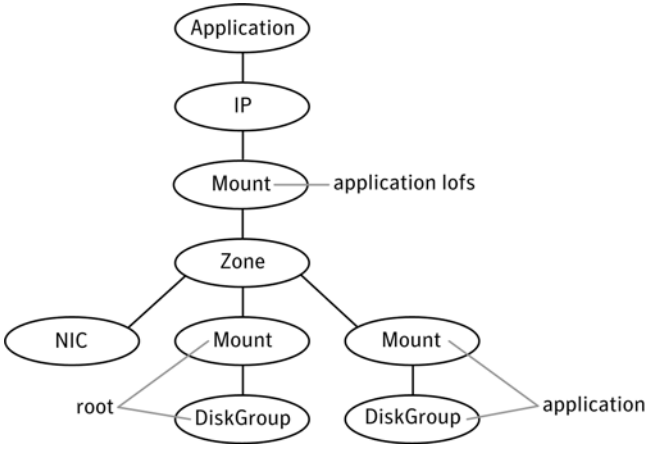
- For IPv6 functionality for NFS, you must use NFS version 4 in order to make the mount reachable. AIX defaults to NFSv3, which does not work across IPv6. Note that NFSv4 requires several configuration steps in the operating system and NFS-related resources in VCS to enable it on the client and the exporting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

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

Support for loopback file system

The Mount agent provides loopback file system support. You can manage the loopback file system as a Mount resource. You can use this loopback support to mount a file system in the global zone and share it in non-global zones. For loopback support, configure the FSType attribute to use a value of lofs.

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



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

```
group loopbacksg (
    SystemList = { sysA = 0, sysB = 1 }
    ContainerInfo@sysA = { Name = zone1, Type = Zone, Enabled = 1 }
    ContainerInfo@sysB = { Name = zone1, Type = Zone, Enabled = 1 }
    )
    Mount lofs_mnt_global_to_local (
                MountPoint = "/export/home/zone1/root/lofs_mnt"
                BlockDevice = "/mnt1/m1"
                FSType = lofs
                )
    Zone z1 (
                )
        Mount base_mnt (
                MountPoint = "/mnt1"
                BlockDevice = "/dev/vx/dsk/tdg/tvol1"
                FSType = vxfs
```

```
                FsckOpt = “-y”
                )
    lofs_mnt_global_to_local requires z1
    lofs_mnt_global_to_local requires base_mnt
```

ZFS file system and pool creation example

If you want to use the Mount resource to monitor the ZFS file system, perform the following steps.

Create the tank storage pool and file system on the disk device c1t0d0 for example.

```
# zpool create tank c1t0d0
```

Create the home file system in tank.

```
# zfs create tank/home
```

Set the value of the MountPoint attribute to legacy.

```
# zfs set mountpoint=legacy tank/home
```

Set the Mount agent's attributes. The following is an example of this configuration's main.cf file.

```
Mount m1 (
    MountPoint = "/mp1"
    BlockDevice = "tank/home"
    FSType = zfs
    MountOpt = rw
    FsckOpt = "-n"
)
```

Enabling second level monitoring for the Mount agent

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

To enable second level monitoring, run the following commands

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

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

Sample configurations

VxFS configuration example

```
Mount mnt-fs1 (
    MountPoint= "/mnt1"
    BlockDevice = "/dev/vx/dsk/mnt-dg1/mnt-vol1"
    FSType = "vxfs"
    FsckOpt = "-n"
    MountOpt = "rw"
)
```

Debug log levels

The Mount agent uses the following debug log levels:

DBG_1, DBG_2, DBG_3, DBG_4, DBG_5

Zpool agent

The Zpool agent monitors ZFS storage pools. It exports ZFS storage pools (which reside on shared storage) from one node. It then imports the pool onto another node as required.

This agent supports Veritas Dynamic Multi-Pathing.

ZFS's automount feature mounts all its file systems by setting the mountpoint property to something other than legacy. To find the value of the mountpoint property, use the `zfs get` command. For example, from the command line for the tank mountpoint, enter:

```
# zfs get mountpoint tank
NAME        PROPERTY       VALUE          SOURCE
tank        mountpoint     /tank          default
```

As another example, to find the value of the mountpoint property for the legacypool storage pool, enter:

```
# zfs get mountpoint legacypool
NAME        PROPERTY       VALUE          SOURCE
tank        mountpoint     legacy         default
```

The Zpool agent checks this property, and checks the ChkZFSMounts attribute to decide whether the mounted file system should be checked in the Zpool agent or not.

When the value of the mountpoint property is one of the following:

- If the value of the mountpoint property is something other than legacy, the agent checks the mount status of the ZFS file systems.
- If the value of the mountpoint property is legacy, then it does not check the file system mount status. The agent assumes that you plan to use Mount resources to manage and monitor the ZFS file systems.

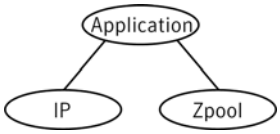
Limitations

The agent does not support the use of logical volumes in ZFS. If ZFS logical volumes are in use in the pool, the pool cannot be exported, even with the `-f` option. Oracle does not recommend the use of logical volumes in ZFS due to performance and reliability issues.

Dependencies

No dependencies exist for the Zpool resource for a pool that has a non-legacy value for its mountpoint property.

Figure 2-10 Sample service group for a Zpool resource



Agent functions

Online	Imports the ZFS storage pool.
Offline	Exports the ZFS storage pool.
Monitor	Checks the online status of the ZFS pool. If the mountpoint property of the ZFS file system is set and its value is not legacy, and the attribute ChkZFSMounts is enabled, then the agent checks if all the ZFS file systems under the same ZFS storage pool are mounted. If the ZFS pool contains a ZFS file system that a non-global zone uses, then you need to import the pool before the zone boots up. After the zone boots up, if the mountpoint property for this ZFS file system that the non-global zone uses is not set to legacy, it mounts after the zone boots up. If you have enabled the ChkZFSMounts in the Zpool resource, delay the check inside the Monitor agent function because the zone resource is not up yet, and the file systems are not mounted until the zone boots up. The Zone resource depends on the Zpool resource for the non-global zone scenario. In this case, you need to provide the ZoneResName attribute, which indicates the name of the Zone resource. When the Zone resource is in an ONLINE state, then ChkZFSMounts starts to check the mount status of the ZFS file system pool that the non-global zone uses.
Clean	Exports the ZFS storage pool forcefully.

State definitions

ONLINE	Reports an ONLINE state when the ZFS file systems that share a common storage pool are mounted, and the zpool command `zpool list -H -o health $Poolname` indicates if the pool is online or not.
OFFLINE	Reports an OFFLINE state when all of the ZFS file systems that share a common storage pool are unmounted. It also reports an OFFLINE state when the zpool command `zpool list -H -o health $Poolname` command's status indicates that the pool is offline.
UNKNOWN	Reports an UNKNOWN state in the following situations: ■ If the status of the storage pool is unavailable or faulted. ■ If the storage pool is online but the path of the mounted file system does not match the path that is specified in the AltRootPath attribute of this agent.

Attributes

Table 2-11 Required attributes

Required attribute	Description
PoolName	The name of the ZFS storage pool name. Type and dimension: string-scalar Default: n/a Example: tank
AltRootPath	Provides the alternate root path that is necessary to prevent the /etc/zfs/zpool.cache file from being populated. Supplying this value keeps a node from importing the ZFS storage pool automatically when it restarts after a crash. Not importing the ZFS storage prevents concurrency violations and file system corruption. If you do not provide a value for the AltRootPath attribute, VCS sets the $AltRootPath to "/" as a workaround. This workaround makes sure that the ZFS command `zpool import -R $AltRootPath $PoolName` does not populate the zpool.cache file. Type and dimension: string-scalar Default: / Example: /mnt
ChkZFSMounts	The ChkZFSMounts attribute enables the check to determine whether all the file systems are properly mounted for that ZFS storage pool when the mountpoint property is not set to legacy. The default value is enabled (set to 1). Type and dimension: boolean-scalar Default: 1

Table 2-12 Optional attributes

Required attribute	Description
ZoneResName	Use the ZoneResName attribute when a non-global zone needs the Zpool resource. In this case, supply the ZoneResName attribute with the name of the Zone resource. Type and dimension: string-scalar Default: n/a Example: zone1

Resource type definition

```
type Zpool (
    static str ArgList[] = { PoolName, AltRootPath, ChkZFSMounts,
    ZoneResName, "ZoneResName:State" }
    str PoolName
    str AltRootPath
    boolean ChkZFSMounts = 1
    str ZoneResName
)
```

Sample configurations

A main.cf example that shows the Zpool agent configuration.

```
include "types.cf"

cluster clus1 (
    UserNames = { admin = dqrJqlQnrMrrPzrLqo,
        z_zone_res = dOMoOTnNMlMSlVPnOT,
        z_dummy_res = bIJbIDiFJeJJhRJdIG }
    Administrators = { admin }
    )

system sysA (
    )

system sysB (
    )

group tstzfs (
    SystemList = { sysA = 0, sysB = 1 }
    AutoStartList = { sysA, sysB }
    ContainerInfo = { Name = z1, Type = Zone, Enabled = 1 }
```

```
Administrators = { z_zone_res }
)

Zone zone_res (
    )

Zpool legacy_res (
    PoolName = legacypool
    ZoneResName = zone_res
    )

Zpool tstzone_res (
    PoolName = tstzonepool
    ZoneResName = zone_res
    )

Zpool zpool_res (
    PoolName = tank
    AltRootPath = "/mytank"
    )

Application custom_app (
    StartProgram = "/mytank/tank/startapp"
    StopProgram = "/mytank/tank/stopapp"
    MonitorProcesses = { "/mytank/tank/mondaemon" }
    )

custom_app requires zpool_res
zone_res requires legacy_res
zone_res requires tstzone_res
```

Debug log levels

The Zpool agent uses following debug log levels:

DBG_1, DBG_2, and DBG_5

Chapter 3

Network agents

This chapter contains the following:

- "About the network agents" on page 79
- "IP agent" on page 82
- "NIC agent" on page 89
- "IPMultiNIC agent" on page 96
- "MultiNICA agent" on page 102
- "About the IPMultiNICB and MultiNICB agents" on page 111
- "IPMultiNICB agent" on page 112
- "MultiNICB agent" on page 119
- "DNS agent" on page 134

About the network agents

Use network agents to provide high availability for networking resources.

Agent comparisons

IP and NIC agents

The IP and NIC agents:

- Monitor a single NIC

IPMultiNIC and MultiNICA agents

The IPMultiNIC and MultiNICA agents:

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

IPMultiNICB and MultiNICB agents

The IPMultiNICB and MultiNICB agents:

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

802.1Q trunking

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

The IP/NIC, IPMultiNIC/MultiNICA, and IPMultiNICB/MultiNICB agents support 802.1Q trunking on Solaris 8, 9 and 10. However, on Solaris 8, only "ce" interfaces can be configured as VLAN interfaces. This is a Sun restriction.

On Solaris 9, the IPMultiNICB and MultiNICB agents works only if Sun patch 116670-04 is installed on the system. No patch is required for the IP and NIC agents and the IPMultiNIC and MultiNICA agents

On Solaris 9 and 10, VLAN is not supported on the Fast Ethernet interfaces. (eg: hme/qfe interfaces).

You need to specify the VLAN interfaces, for example: bge20001 , bge30001, as the base interfaces in the device list in the main.cf file. You also must make sure that the IP addresses that are assigned to the interfaces of a particular VLAN are in the same subnet.

Link aggregation support

The link aggregation feature aggregates multiple network interfaces so that they appear as a single interface. For example, you can combine bge0 and bge1 and name the combined interface aggr100.

You can use the NIC, MultiNICA, or MultiNICB agents to monitor an aggregated interface. You can use the IP, IPMultiNIC, or IPMultiNICB agent respectively to configure and monitor an IP address on the aggregated interface.

All VCS networking agents support link aggregation. However, VCS has no control over the local adapter swapping performed by the link aggregation module. For guidelines on creating and managing link aggregations, refer to the topic *Overview of Link Aggregations* in the *System Administration Guide: IP Services* guide available on the Oracle Solaris documentation website.

Note: After you set up an aggregated interface, the constituting interfaces are no longer configurable. Hence, you must specify aggregated interfaces while configuring the Device attribute of IP, NIC, MultiNICA, or MultiNICB resources.

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.

The ContainerName and ContainerType attributes are deprecated.

This agent is zone-aware. The ContainerOpts resource type attribute for this type has a default value of 0 for RunInContainer and a default value of 1 for PassCInfo. Symantec recommends that you do not change these values. Refer to the *Storage Foundation High Availability Virtualization Guide* for more information.

High availability fire drill

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

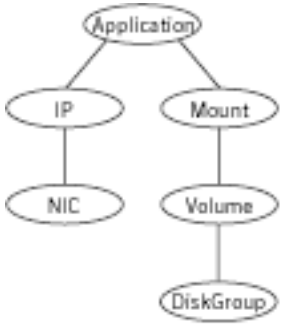
- Checks for the existence of a route to the IP from the specified NIC
- Checks for the existence of the interface configured in the IP and NIC resources
- Checks for the validity of the configured PrefixLen attribute value for IPv6 configuration

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

Dependencies

IP resources depend on NIC resources.

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



Agent functions

Online	Configures the IP address to the NIC. Checks if another system is using the IP address. Uses the `ifconfig` command to set the IP address on a unique alias on the interface.
Action	The various functions of the action agent are as follows: ■ route.vfd Checks for the existence of a route to the IP from the specified NIC. Checks for the validity of the configured PrefixLen attribute value for IPv6 addresses. ■ device.vfd Checks for the existence of the interface configured in the Device attribute.
Offline	Brings down the IP address that is specified in the Address attribute.
Monitor	Monitors the interface to test if the IP address that is associated with the interface is alive.
Clean	Brings down the IP address that is associated with the specified interface.

State definitions

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

Attributes

Table 3-1 Required attributes

Required attribute	Description
Address	A virtual IP address that is associated with the interface. Note that the address you specify must not be the same as the configured physical IP address, but should be on the same network. Type and dimension: string-scalar Examples: IPv4: "192.203.47.61" IPv6: "2001::10"
Device	The name of the NIC device that is associated with the IP address. Requires the device name without an alias. Type and dimension: string-scalar Example: "le0"
PrefixLen	Required to use the IPv6 protocol. See "PrefixLen" on page 87.

Table 3-2 Optional attributes

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

Table 3-2 Optional attributes

Optional attribute	Description
IfconfigTwice	Causes an IP address to be configured twice using an ifconfig up-down-up sequence. Increases the probability of gratuitous ARP requests (generated by `ifconfig up`) to reach clients. Type and dimension: integer-scalar Default: 0
NetMask	The subnet mask that is associated with the IP address of the resource. Specify the value of the netmask in decimal (base 10) or hexadecimal (base 16). Configure this attribute only if the IP address is an IPv4 address. Symantec recommends that you specify a netmask for each virtual interface. Type and dimension: string-scalar Default: + If you do not specify the netmask in the `ifconfig` command, the agent uses a default netmask that is based on the contents of the /etc/netmasks for a given address range. Example: "255.255.248.0"
Options	Options for the `ifconfig` command. Type and dimension: string-scalar Example: "failover"

Table 3-2 Optional attributes

Optional attribute	Description
RouteOptions	Specifies the routing options that are passed to the `route add` command when the agent configures an interface. The RouteOptions attribute value is generally formed like this: `"destination gateway metric"`. For details about the `route` command, refer to the man page for your operating system. When the value of this string is null, the agent does not add routes. Type and dimension: string-scalar Example: "192.100.201.0 192.100.13.7" In this example, the agent executes the `"route add 192.100.201.0 192.100.13.7"` command when it configures an interface.
PrefixLen	This is the prefix for the IPv6 address represented as the CIDR value. When you use the IPv6 protocol, you must configure values for this attribute and the corresponding NIC agent's Device and Protocol attributes. Type-dimension: integer-scalar Range: 1- 128 Example: 64
ExclusiveIPZone	Specifies that a resource is configured for an exclusive IP zone. Set this value to 1 if resource is configured for exclusive IP zone. When set to 1, it requires a valid ContainerInfo to be configured in the service group. Type-dimension: boolean-scalar Default: 0

Resource type definition

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

```
    str Device
    str Address
    str NetMask
    str Options
    int ArpDelay = 1
    int IfconfigTwice
    str RouteOptions
    int PrefixLen
    boolean ExclusiveIPZone = 0
)
```

Sample configurations

Configuration 1

```
IP IP_192_203_47_61 (
    Device = le0
    Address = "192.203.47.61"
    )
```

NetMask in decimal (base 10)

```
IP IP_192_203_47_61 (
    Device = le0
    Address = "192.203.47.61"
    NetMask = "255.255.248.0"
    )
```

Configuration of NetMask in hexadecimal (base 16)

```
IP IP_192_203_47_61 (
    Device = le0
    Address = "192.203.47.61"
    NetMask = "0xfffff800"
    )
```

Debug log levels

The IP agent uses the following debug log levels:

DBG_1, DBG_3, 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.

The ContainerName and ContainerType attributes are deprecated.

This agent is zone-aware. The ContainerOpts resource type attribute for this type has a default value of 0 for RunInContainer and a default value of 1 for PassCInfo. Symantec recommends that you do not change these values. Refer to the *Storage Foundation High Availability Virtualization Guide* for more information.

High availability fire drill

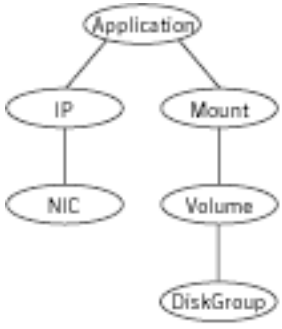
The high availability fire drill detects discrepancies between the VCS configuration and the underlying infrastructure on a node. These discrepancies might prevent a service group from going online on a specific node. For NIC resources, the high availability fire drill checks for the existence of the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

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

Dependencies

The NIC resource does not depend on any other resources.

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



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

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

Before you use this agent:

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

Agent functions

Monitor	Tests the network card and network link. Pings the network hosts or broadcast address of the interface to generate traffic on the network. Counts the number of packets passing through the device before and after the address is pinged. If the count decreases or remains the same, the resource is marked FAULTED. If the NetworkHosts list is empty, or the ping test fails, the agent sends a ping to the device's broadcast address to generate network traffic. The agent checks for any response to the broadcast request. If there is no reply to the broadcast ping, the resource faults.
Action	■ clearNICFaultInZone This action entry point clears the fault on the NIC device that is configured for Exclusive IP zone.

State definitions

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

Attributes

Table 3-3 Required attributes

Required attribute	Description
Device	Name of the NIC that you want to monitor. Type and dimension: string-scalar Example: "le0"
Protocol	Required to use the IPv6 protocol. See "Protocol" on page 92.

Table 3-4 Optional attributes

Optional attribute	Description
NetworkHosts	List of hosts on the network that are pinged to determine if the network connection is alive. You can use this attribute to help to save network capacity and reduce monitor time. Symantec recommends that you use the outgoing gateway routers for this value. Enter the IP address of the host, instead of the host name, to prevent the monitor from timing out. DNS causes the ping to hang. If more than one network host is listed, the monitor returns ONLINE if at least one of the hosts is alive. If an invalid network host address is specified or if there is mismatch in protocol of network host and Protocol attribute of the resource, the resource enters an UNKNOWN state. If you do not specify network hosts, the monitor tests the NIC by sending pings to the broadcast address on the NIC. Type and dimension: string-vector Example: "166.96.15.22", "166.97.1.2"
NetworkType	Type of network. VCS supports only Ethernet. Type and dimension: string-scalar Default: "ether"

Table 3-4 Optional attributes

Optional attribute	Description
PingOptimize	Number of monitor cycles to detect if a configured interface is inactive. Use PingOptimize when you have not specified the NetworkHosts attribute. A value of 1 optimizes broadcast pings and requires two monitor cycles. A value of 0 performs a broadcast ping during each monitor cycle and detects the inactive interface within the cycle. Type and dimension: integer-scalar Default: 1
Protocol	The type of IP protocol (IPv4 or IPv6) that you want to use with the agent. When you use the IPv6 protocol, you must configure values for the Device attribute and the corresponding IP agent's PrefixLen attribute. Type-dimension: string-scalar Default: IPv4 Example: IPv6
ExclusiveIPZone	Specifies that a resource is configured for an exclusive IP zone. Set this value to 1 if a resource is configured for exclusive IP zone. When set to 1, it requires a valid ContainerInfo to be configured in the service group. Type-dimension: boolean-scalar Default: 0

Resource type definition

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

```
    str NetworkHosts[]
    str Protocol = IPv4
    str NetworkType
    boolean ExclusiveIPZone = 0
)
```

Sample configurations

Configuration without network hosts (using default ping mechanism)

```
NIC groupx_le0 (
    Device = le0
    PingOptimize = 1
    )
```

Configuration with network hosts

```
NIC groupx_le0 (
    Device = le0
    NetworkHosts = { "166.93.2.1", "166.99.1.2" }
    )
```

IPv6 configuration

The following is a basic configuration for IPv6 with IP and NIC resources. In the following sample, *nic_value* represents the base NIC value for the platform (for example, bge0).

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

    NIC nic_resource (
        Device@sysA = bge0
        Device@sysB = bge1
        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 = bge0
        Device@sysB = bge1
        Address = "2001:db8:c18:2:214:4fff:fe96:102"
        PrefixLen = 64
        )

    Proxy proxy_resource (
        TargetResName = nic_resource
        )

    ip_resource requires proxy_resource
```

Exclusive IP Zone configuration

Following is the configuration example for Exclusive IP zone with NIC and IP resources. In the following sample, *nic_value* represents the base NIC name for the platform (for example, bge0) and *zone_name* is the name of the exclusive IP zone. (For more details about Zone resource configuration, refer to the Zone agent section.)

```
group grp1 (
        SystemList = { sysA = 0 }
        ContainerInfo @sysA = { Name = zone_name, Type = Zone,
Enabled = 1 }
        AutoStartList = { sysA }
        Administrators = { z_zone_res_sysA }
        )

        IP ip_res (
                Device = nic_value
                Address = "166.93.3.10"
                NetMask = "255.255.255.0"
                ExclusiveIPZone = 1
                )

        NIC nic_res (
                Device = nic_value
                NetworkHosts = { "166.93.3.1" }
                ExclusiveIPZone = 1
                )

        Zone zone_res (
                )

        ip_res requires nic_res
        ip_res requires zone_res
```

Note that whenever a fault is detected for a NIC resource configured in an exclusive IP zone, perform the following steps to clear the fault.

1. Repair the device configured with NIC resource. Verify that the device is healthy (check for cable connectivity, network connectivity, and so on).
2. If the state of the exclusive IP zone on the system on which the NIC was faulted is:
   a. Running: No action is required, and the next NIC monitor cycle will clear the fault after detecting the healthy NIC device.
   b. NOT running: Clear the fault on the NIC device by invoking 'clearNICFaultInZone' action entry point for the NIC resource as follows:

```
# hares -action nic_res clearNICFaultInZone -sys sysA
```

Debug log levels

The NIC agent uses the following debug log levels:

DBG_1, DBG_5

IPMultiNIC agent

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

The IPMultiNIC and MultiNICA agents supports IPv4 and IPv6.

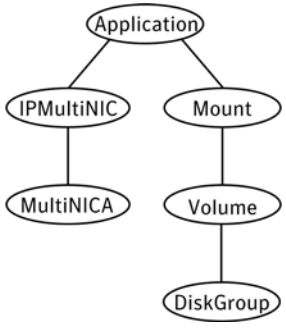
The ContainerName and ContainerType attributes are deprecated.

This agent is zone-aware. The ContainerOpts resource type attribute for this type has a default value of 0 for RunInContainer and a default value of 1 for PassCInfo. Symantec recommends that you do not change these values. Refer to the *Storage Foundation High Availability Virtualization Guide* for more information.

Dependencies

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

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



Agent functions

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

State definitions

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

Attributes

Table 3-5 Required attributes

Required attribute	Description
Address	Virtual IP address assigned to the active NIC. Type and dimension: string-scalar Examples: IPv4: "10.128.10.14" IPv6: "2001:DB8::"
MultiNICResName	Name of the associated MultiNICA resource that determines the active NIC. Type and dimension: string-scalar Example: "mnic"
PrefixLen	Required to use the IPv6 protocol. See "PrefixLen" on page 99.

Table 3-6 Optional attributes

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

Table 3-6 Optional attributes

Optional attribute	Description
NetMask	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). Symantec recommends that you specify a netmask for each virtual interface. Type and dimension: string-scalar Default: + If you do not specify the netmask in the `ifconfig` command, the agent uses a default netmask. The default netmask is based on the contents of the /etc/netmasks for a given address range. Example: "255.255.248.0"
Options	The `ifconfig` command options for the virtual IP address. Type and dimension: string-scalar Example: "failover"
PrefixLen	Specifies the prefix for the IPv6 address represented as the CIDR value. When you use the IPv6 protocol, you must configure values for this attribute and the MultiNICA agent's Device and Protocol attributes. Type-dimension: integer-scalar Range: 1 - 128 Example: 64

Note: On Solaris systems, Symantec recommends that you set the RestartLimit for IPMultiNIC resources to a greater-than-zero value. Setting this value helps to prevent the spurious faulting of IPMultiNIC resources during local failovers of MultiNICA. A local failover is an interface-to- interface failover of MultiNICA. See the *Veritas Cluster Server Administrator's Guide* for more information.

Resource type definition

```
type IPMultiNIC (
```

```
    static str ArgList[] = { "MultiNICResName:Device", Address,
    NetMask, "MultiNICResName:ArpDelay", Options,
    "MultiNICResName:Probed", MultiNICResName, IfconfigTwice,
    PrefixLen }
    static int ContainerOpts{} = { RunInContainer=0, PassCInfo=1 }
    static int MonitorTimeout = 120
    str Address
    str NetMask
    str Options
    str MultiNICResName
    int IfconfigTwice
    int PrefixLen
)
```

Sample configuration: IPMultiNIC and MultiNICA

Refer to the MultiNICA agent for more information.

```
group grp1 (
    SystemList = { sysa = 0, sysb = 1 }
    AutoStartList = { sysa }
    )
    MultiNICA mnic (
        Device@sysa = { le0 = "10.128.8.42", qfe3 = "10.128.8.42" }
        Device@sysb = { le0 = "10.128.8.43", qfe3 = "10.128.8.43" }
        NetMask = "255.255.255.0"
        ArpDelay = 5
        Options = "failover"
        )
    IPMultiNIC ip1 (
        Address = "10.128.10.14"
        NetMask = "255.255.255.0"
        MultiNICResName = mnic
        Options = "failover"
        )
ip1 requires mnic
group grp2 (
    SystemList = { sysa = 0, sysb = 1 }
    AutoStartList = { sysa }
    )
        IPMultiNIC ip2 (
        Address = "10.128.9.4"
        NetMask = "255.255.255.0"
        MultiNICResName = mnic
        Options = "failover"
        )
    Proxy proxy (
        TargetResName = mnic
        )
ip2 requires proxy
```

Debug log levels

The IPMultiNIC agent uses the following debug log levels:

DBG_1

MultiNICA agent

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

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

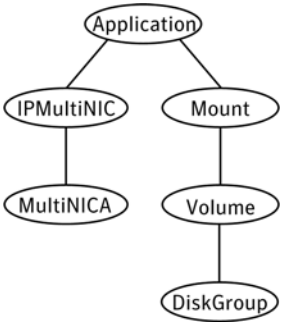
For important information on this agent, refer to:

See "MultiNICA notes" on page 108.

Dependencies

The MultiNICA resource does not depend on any other resources.

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



Agent function

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

State definitions

ONLINE	Indicates that one or more of the network interfaces listed in the Device attribute of the resource is in working condition.
FAULTED	Indicates that all of the network interfaces listed in the Device attribute failed.
UNKNOWN	Indicates that the agent cannot determine the state of the network interfaces that are specified in the Device attribute. This state may be due to incorrect configuration.

Attributes

Table 3-7 Required attributes

Required attribute	Description
Device	List of interfaces and their base IP addresses. Type and dimension: string-association Example: { le0 = "10.128.8.42", qfe3 = "10.128.8.42" }
Protocol	Required to use the IPv6 protocol. See "Protocol" on page 107.

Table 3-8 Optional attributes

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

Table 3-8 Optional attributes

Optional attribute	Description
HandshakeInterval	Computes the maximum number of tries the agent makes either to: ■ ping a host (listed in the NetworkHosts attribute) when it fails over to a new NIC, or ■ to ping the default broadcast address (depending on the attribute configured) when it fails over to a new NIC. If the value of the RetestInterval attribute is five (default), each try takes about 10 seconds. To prevent spurious failovers, the agent must try to contact a host on the network several times before it marks a NIC as FAULTED. Increased values result in longer failover times, whether between the NICs or from system to system in the case of FAULTED NICs. Type and dimension: integer-scalar Default: 20 This value is the equivalent to two tries (20/10).
IfconfigTwice	Causes an IP address to be configured twice, using an `ifconfig up-down-up` sequence. Increases the probability of gratuitous ARP requests (caused by `ifconfig up`) to reach clients. Type and dimension: integer-scalar
NetMask	Netmask for the base IP address. Specify the value of NetMask in decimal (base 10). **Note:** Symantec recommends that you specify a netmask for each virtual interface. Type and dimension: string-scalar Default: + Example: "255.255.255.0"

Table 3-8 Optional attributes

Optional attribute	Description
NetworkHosts	The list of hosts on the network that are pinged to determine if the network connection is alive. Enter the IP address of the host, instead of the host name, to prevent the monitor from timing out—DNS causes the ping to hang. If this attribute is unspecified, the monitor tests the NIC by pinging the broadcast address on the NIC. If more than one network host is listed, the monitor returns online if at least one of the hosts is alive. If an invalid network host address is specified or if there is a mismatch in protocol of the network host and Protocol attribute of resource, the resource enters an UNKNOWN state. Type and dimension: string-vector Example: "128.93.2.1", "128.97.1.2"
Options	The `ifconfig` options for the base IP address. Type and dimension: string-scalar Example: "failover"
PingOptimize	Number of monitor cycles to detect if the configured interface is inactive. A value of 1 optimizes broadcast pings and requires two monitor cycles. A value of 0 performs a broadcast ping each monitor cycle and detects the inactive interface within the cycle. Type and dimension: integer-scalar Default: 1
RetestInterval	Number of seconds to sleep between re-tests of a newly configured interface. A lower value results in faster local (interface-to-interface) failover. Type and dimension: integer-scalar Default: 5

Table 3-8 Optional attributes

Optional attribute	Description
RouteOptions	String to add a route when configuring an interface. Use only when configuring the local host as the default gateway. The string contains destination gateway metric. No routes are added if this string is set to NULL. Type and dimension: string-scalar Example: "default 166.98.16.103 0"
Protocol	Specifies the type of IP protocol (IPv4 or IPv6) that you want to use with the agent. When you use the IPv6 protocol, you must configure values for this attribute, the Device attribute, and the corresponding IPMultiNIC agent's PrefixLen attribute. Type-dimension: string-scalar Default: IPv4 Example: IPv6

Resource type definition

```
type MultiNICA (
    static str ArgList[] = { Device, NetMask, ArpDelay,
    RetestInterval, Options, RouteOptions, PingOptimize,
    MonitorOnly, IfconfigTwice, HandshakeInterval, NetworkHosts,
    Protocol }
    static int OfflineMonitorInterval = 60
    static int MonitorTimeout = 300
    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
)
```

MultiNICA notes

- If all NICs configured in the Device attribute are down, the MultiNICA agent faults the resource after a two-three minute interval. This delay occurs because the MultiNICA agent tests the failed NIC several times before it marks the resource OFFLINE. Failover logs record a detailed description of the events.
- The engine log is in `/var/VRTSvcs/log/engine_A.log`.
- The MultiNICA agent supports only one active NIC on one IP subnet; the agent does not work with multiple active NICs on the same subnet.
  - On Solaris, for example, you have two active NICs, hme0 (10.128.2.5) and qfe0 (10.128.2.8). You configure a third NIC, qfe1, as the backup NIC to hme0. The agent does not fail over from hme0 to qfe1 because all ping tests are redirected through qfe0 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

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

For example, if the default gateway and hme0 are both set to 10.128.8.42, the output of the `netstat -rn` command resembles:

```
Destination      Gateway           Flags Ref   Use    Interface
---------------- ----------------- ----- ----- ------ ---------
10.0.0.0         10.128.8.42         U    1     2408   hme0
224.0.0.0        10.128.8.42         U    1     0      hme0
default          10.128.8.42         UG   1     2402   hme0
127.0.0.1        127.0.0.1           UH   54    44249  lo0
```

If the RouteOptions attribute is not set and hme0 fails, the MultiNICA agent migrates the base IP address to another NIC (such as qfe0). The default route is no longer configured because it was associated with hme0. The display resembles:

```
Destination      Gateway           Flags Ref   Use    Interface
---------------- ----------------- ----- ----- ------ ---------
10.0.0.0         10.128.8.42         U    1    2408    qfe0
224.0.0.0        10.128.8.42         U    1    0       qfe0
127.0.0.1        127.0.0.1           UH   54   44249   lo0
```

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

```
RouteOptions@sysa = "default 10.128.8.42 0"
RouteOptions@sysb = "default 10.128.8.43 0"
```

Sample configurations

MultiNICA and IPMultiNIC

In the following example, two nodes, sysa and sysb, each have a pair of network interfaces, le0 and qfe3. In this example, the two interfaces, le0 and qfe3, have the same base, or physical, IP address. Note the lines beginning Device@sysa and Device@sysb; the use of different physical addresses shows how to localize an attribute for a particular host.

The MultiNICA resource fails over only the physical IP address to the backup NIC during a failure. The IPMultiNIC agent configures the logical IP addresses. The resources ip1 and ip2, shown in the following example, have the Address attribute that contains the logical IP address. If a NIC fails on sysa, the physical IP address and the two logical IP addresses fails over from le0 to qfe3. If qfe3 fails, the address fails back to le0 if le0 is reconnected.

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

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

```
group grp1 (
    SystemList = { sysa = 0, sysb = 1 }
    AutoStartList = { sysa }
    )
    MultiNICA mnic (
        Device@sysa = { le0 = "10.128.8.42", qfe3 = "10.128.8.42" }
        Device@sysb = { le0 = "10.128.8.43", qfe3 = "10.128.8.43" }
        NetMask = "255.255.255.0"
        ArpDelay = 5
        Options = "failover"
        )

    IPMultiNIC ip1 (
        Address = "10.128.10.14"
        NetMask = "255.255.255.0"
        MultiNICResName = mnic
```

```
        Options = "failover"
        )

ip1 requires mnic

group grp2 (
    SystemList = { sysa = 0, sysb = 1 }
    AutoStartList = { sysa }
    )
IPMultiNIC ip2 (
        Address = "10.128.9.4"
        NetMask = "255.255.255.0"
        MultiNICResName = mnic
        Options = "failover"
        )
    Proxy proxy (
        TargetResName = mnic
        )

ip2 requires proxy
```

IPv6 configuration

The following is a basic configuration for IPv6.

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

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

    MultiNICA mnica_res (
        Device @sysA = { e1000g1 = "fe80::214:4fff:fe96:ae0a",
e1000g3 = "fe80::214:4fff:fe96:ae0b" }
        Device @sysB = { e1000g1 = "fe80::214:4fff:fe96:ae1a",
e1000g3 = "fe80::214:4fff:fe96:ae1b" }
        Protocol = IPv6
        )

    ipmnic_res requires mnica_res
```

Debug log levels

The MultiNICA agent uses the following debug log levels:

DBG_1

About the IPMultiNICB and MultiNICB agents

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

Checklist to ensure the proper operation of MultiNICB

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

- Each interface must have a unique MAC address.
- A MultiNICB resource controls all the interfaces on one IP subnet.
- At boot time, you must configure and connect all the interfaces that are under the MultiNICB resource and give them test IP addresses.
- All test IP addresses for the MultiNICB resource must belong to the same subnet as the virtual IP address.
- Reserve the base IP addresses, which the agent uses to test the link status, for use by the agent. These IP addresses do not get failed over.
- The IgnoreLinkStatus attribute is set to 1 (default) when using trunked interfaces.
- If you specify the NetworkHosts attribute, then that host must be on the same subnet as the other IP addresses for the MultiNICB resource.
- Test IP addresses have "nofailover" and "deprecated" flags set at boot time.
- /etc/default/mpathd/ has TRACK_INTERFACES_ONLY_WITH_GROUPS=yes.
- If you are not using Solaris in.mpathd, all MultiNICB resources on the system have the UseMpathd attribute set to 0 (default). You cannot run in.mpathd on this system.
- If you are using Solaris in.mpathd, all MultiNICB resources on the system have the UseMpathd attribute set to 1.

IPMultiNICB agent

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

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

For the MultiNICB and IPMultiNICB agents, VCS supports Oracle trunking.

The ContainerName and ContainerType attributes are deprecated.

This agent is zone-aware. The ContainerOpts resource type attribute for this type has a default value of 0 for RunInContainer and a default value of 1 for PassCInfo. Symantec recommends that you do not change these values. Refer to the *Storage Foundation High Availability Virtualization Guide* for more information.

For the MultiNICB and IPMultiNICB agents, VCS supports IPv4 and IPv6.

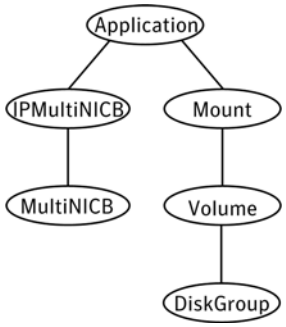
The value of the MonitorInterval attribute for the MultiNICB type must be less than its value for the IPMultiNICB type. The IPMultiNICB agent relies on the MultiNICB agent to accurately report the state of the NICs. If the value of the MonitorInterval of the IPMultiNICB agent is less than the interval for the MultiNICB agent, then in some monitor cycles the potential exists for the MultiNICB agent to provide the IPMultiNICB agent stale information.

The default value for the MonitorInterval attribute for IPMultiNICB is 30 seconds, and the minimum and default value for MultiNICB is 10 seconds.

Dependencies

IPMultiNICB resources depend on MultiNICB resources.

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



Requirements for IPMultiNICB

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

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

Agent functions

Online	Finds a working interface with the appropriate interface alias or interface name, and configures the logical IP address on it.
Offline	Removes the logical IP address.
Clean	Removes the logical IP address.
Monitor	If the logical IP address is not configured as an alias on one of the working interfaces under a corresponding MultiNICB resource, monitor returns OFFLINE. If the current interface fails, the agent fails over the logical IP address. It fails over the logical IP address to the next available working interface that is within the MultiNICB resource on the same node. If no working interfaces are available then monitor returns OFFLINE.

State definitions

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

Attributes

Table 3-9 Required attributes

Required attribute	Description
Address	The logical IP address that the IPMultiNICB resource must handle. This IP address must be different than the base or test IP addresses in the MultiNICB resource. Type and dimension: string-scalar Example: "10.112.10.15"
BaseResName	Name of MultiNICB resource from which the IPMultiNICB resource gets a list of working interfaces. The logical IP address is placed on the physical interfaces according to the device number information. Type and dimension: string-scalar Example: "gnic_n"
NetMask	The netmask that is associated with the logical IP address. Type and dimension: string-scalar Example: "255.255.255.0"
PrefixLen	Required to use the IPv6 protocol. See "PrefixLen" on page 117.

Table 3-10 Optional attributes

Optional attribute	Description
DeviceChoice	Indicates the preferred NIC where you want to bring the logical IP address online. Specify the device name or NIC alias as determined in the Device attribute of the MultiNICB resource. Type and dimension: string-scalar Default: 0 Examples: "qfe0" and "1"
RouteOptions	Specifies the routing options that are passed to the `route add` command when the agent configures an interface. The RouteOptions attribute value is generally formed like this: "*`destination gateway metric`*". For details about the `route` command, refer to the man page for your operating system. When the value of this string is null, the agent does not add routes. Type and dimension: string-scalar Example: "192.100.201.0 192.100.13.7" In this example, the agent executes the `"route add 192.100.201.0 192.100.13.7"` command when it configures an interface.
IgnoreMultiNICB Failure	Set this value to ignore a MultiNICB resource failure when all configured interfaces fail. A value of 1 for this attribute causes the IPMultiNICB agent to ignore the failure that its underlying MultiNICB resource detects. A value of 0 for this attribute causes the IPMultiNICB agent to detect network failure. When the value of this attribute is 1, the value for the MultiNICB LinkTestRatio attribute cannot be 0. Type and dimension: integer-scalar Default: 0

Table 3-10 Optional attributes

Optional attribute	Description
Options	Options for the `ifconfig` command. Type and dimension: string-scalar Example: "failover"
PrefixLen	This is the prefix for the IPv6 address represented as the CIDR value. When you use the IPv6 protocol, you must configure values for this attribute and the corresponding MultiNICB agent's Device and Protocol attributes. Type-dimension: integer-scalar Range: 1 - 128 Example: 64

Note: The value of the ToleranceLimit static attribute is 1. A value of 1 avoids spurious agent faults in the multi-pathing mode while Oracle's mpathd daemon migrates the IP address from one interface to another.
Due to the change in the ToleranceLimit attribute, the value of the MonitorInterval static attribute is now 30 seconds. The 30-second value means that the agent tries to online the resource twice a minute. This value ensures that the overall fault detection time is still 60 seconds.

Resource type definition

```
type IPMultiNICB (
    static int ToleranceLimit = 1
    static int MonitorInterval = 30
    static int OnlineRetryLimit=1
    static str ArgList[] = { BaseResName, Address, NetMask,
    DeviceChoice, RouteOptions, PrefixLen, IgnoreMultiNICBFailure,
    "BaseResName:Protocol", Options }
    static int ContainerOpts{} = { RunInContainer=0, PassCInfo=1 }
    str BaseResName
    str Address
    str NetMask
    str DeviceChoice = 0
    str RouteOptions
    str Options
    int PrefixLen
    int IgnoreMultiNICBFailure = 0
```

```
)
```

Manually migrating a logical IP address

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

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

```
# haipswitch -s MultiNICB_resname
```

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

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

Sample configurations

Other sample configurations for IPMultiNICB and MultiNICB

Refer to the sample configurations in the MultiNICB agent.

Debug log levels

The IPMultiNICB agent uses the following debug log levels:

DBG_1, DBG_2, DBG_3, DBG_4, DBG_5

MultiNICB agent

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

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

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.

For the MultiNICB and IPMultiNICB agents, VCS supports Oracle trunking.

For the MultiNICB and IPMultiNICB agents, VCS supports IPv4 and IPv6.

The value of the MonitorInterval attribute for the MultiNICB type must be less than its value for the IPMultiNICB type. The IPMultiNICB agent relies on the MultiNICB agent to accurately report the state of the NICs. If the value of the MonitorInterval of the IPMultiNICB agent is less than the interval for the MultiNICB agent, then in some monitor cycles the potential exists for the MultiNICB agent to provide the IPMultiNICB agent stale information.

The default value for the MonitorInterval attribute for IPMultiNICB is 30 seconds, and the minimum and default value for MultiNICB is 10 seconds.

Base and Multi-Pathing modes

You can use the MultiNICB agent in one of two modes. They are:

- Base mode
- Multi-Pathing mode

See "Solaris operating modes: Base and Multi-Pathing" on page 128.

Oracle trunking

You can configure MultiNICB for use with a single trunk head or multiple trunk heads. You need to set the value of the IgnoreLinkStatus attribute to 1. You must also ensure that all interfaces that belong to the same MultiNICB resource are in the same subnet.

The haping utility

Use the haping utility (/opt/VRTSvcs/bin/MultiNICB/haping) to test each NIC before you configure the MultiNICB resource. This utility takes the NIC interface

as an argument. You can use this utility to perform a link test, a broadcast ping, or to ping a specific remote host. Symantec recommends that the administrator perform a test ping with the remote host before adding it to the NetworkHosts parameter. Note that the remote host should be on the same network as the interface from which you are performing the test ping.

Some examples of the command syntax are as follows:

Examples for Solaris

Link test only on interface bge0:

```
haping -l bge0
```

Ping a remote host 10.10.10.10 from interface bge0:

```
haping -g 10.10.10.10 bge0
```

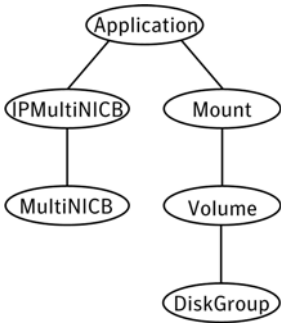
Ping a remote IPv6 host from interface bge0:

```
haping -g fe80::1 bge0
```

Dependencies

The MultiNICB resource does not depend on any other resources.

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



Agent functions

Open	Allocates an internal structure to store information about the resource.
Close	Frees the internal structure that is used to store information about the resource.

Monitor	Checks the status of each physical interface. Writes the status information to the export information file for IPMultiNICB resources to read it. Performs a failover. Performs failback if the value of the Failback attribute is 1.

State definitions

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

Attributes

Table 3-11 Required attributes

Required attribute	Description
Device	List of NICs that you want under MultiNICB control. For IPv4, you must manually configure the test IP addresses on each NIC. This address must be the base IP address (the first address) on that NIC. For IPv6, VCS uses link local addresses as the test IP addresses. You do not need to configure the test IP address. When you use the IPv6 protocol, you must configure values for this attribute, the Protocol attribute, and the corresponding IPMultiNICB agent's PrefixLen attribute. Type and dimension: string-association Examples: Device = { "qfe0" = 0, "qfe1" = 1 }
Protocol	Required to use the IPv6 protocol. See "Protocol" on page 123.

Optional attributes for Base and Mpathd modes

Table 3-12 Optional attributes for Base and Mpathd modes

Optional attribute	Description
DefaultRouter	This attribute is the IP address of the default router on the subnet. If you specify this value, 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. Type and dimension: string-scalar Default: 0.0.0.0 Example: "192.1.0.1"
GroupName	The GroupName attribute is the name of the IPMP group that you want to assign to the interfaces under the control of the agent. The name's length should not exceed 31 characters. If this attribute is not specified, the MultiNICB resource name is configured as IPMP group name. Type and dimension: string-scalar Example: "IPMPgrp1"
MpathdCommand	This value is the path to the mpathd executable. Use MpathdCommand to kill or restart mpathd. See the UseMpathd attribute for details. Type and dimension: string-scalar Default: /usr/lib/inet/in.mpathd

Table 3-12 Optional attributes for Base and Mpathd modes

Optional attribute	Description
UseMpathd	The legal values for this attribute are 0 and 1. All the MultiNICB resources on one system must have the same value for this attribute. See "Solaris operating modes: Base and Multi-Pathing" on page 128. If the value of the attribute is 0, in.mpathd is automatically killed on that system. For more information about mpathd, refer to the Oracle documentation. If the value of the attribute is 1, MultiNICB assumes that mpathd (in.mpathd) is running. This value restarts mpathd if it is not running already. Type and dimension: integer-scalar Default: 0
Protocol	Specifies the type of IP protocol (IPv4 or IPv6) that you want to use with the agent. When you use the IPv6 protocol, you must configure values for this attribute, the Device attribute, and the corresponding IPMultiNICB agent's PrefixLen attribute. Type-dimension: string-scalar Default: IPv4 Example: IPv6

Optional attributes for Base mode

Table 3-13 Optional attributes for Base mode

Optional attribute	Description
Failback	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. Type and dimension: integer-scalar Default: 0

Table 3-13 Optional attributes for Base mode

Optional attribute	Description
IgnoreLinkStatus	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. When you use trunked interfaces, you must set the value of this attribute to 1. Otherwise set it to 0. Type and dimension: integer-scalar Default: 1
LinkTestRatio	This attribute is the ratio of: ■ The monitor cycles in which the agent tests the interfaces by sending packets, to ■ The total monitor cycles 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. If the value of the attribute is 1, packets are sent during every monitor cycle. If the value of the attribute is 0, packets are never sent during a monitor cycle. Do not set the value of this attribute to 0 when its corresponding IPMultiNICB resource's IgnoreMultiNICBFailure attribute has a value of 1. Type and dimension: integer-scalar Default: 1 Example: 3 In this example, if monitor entry-point invoking is numbered as 1, 2, 3, 4, 5, 6, ..., the actual packet send test is done at 3, 6, etc. monitor agent functions. For LinkTestRatio=4, the packet send test is done at 4, 8, etc., monitor agent functions.

Table 3-13 Optional attributes for Base mode

Optional attribute	Description
NetworkHosts	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). 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. Type and dimension: string-vector Example: "192.1.0.1"
NetworkTimeout	Timeout for ARP and ICMP packets in milliseconds. MultiNICB waits for response to ICMP and ARP packets only during this time period. 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. Type and dimension: integer-scalar Default: 100
NoBroadcast	If the value of the attribute is 1, NoBroadcast prevents MultiNICB from sending broadcast ICMP packets. Note that MultiNICB can still send ARP requests. If NetworkHosts are not specified and NoBroadcast is set to 1, the MultiNICB agent cannot function properly. **Note:** Symantec does not recommend setting the value of NoBroadcast to 1. Type and dimension: integer-scalar Default: 0

Table 3-13 Optional attributes for Base mode

Optional attribute	Description
OfflineTestRepeatCount	Number of times the test is repeated if the interface status changes from UP to DOWN. For every repetition of the test, the next 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. Type and dimension: integer-scalar Default: 3
OnlineTestRepeatCount	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. Type and dimension: integer-scalar Default: 3

Optional attributes for Multi-Pathing mode

Table 3-14 Optional attributes for Multi-Pathing mode

Optional attribute	Description
ConfigCheck	If the value of the attribute is 1, the MultiNICB agent checks for: ■ All specified physical interfaces are in the same IP subnet and group, and have "DEPRECATED" and "NOFAILOVER" flags set on them. ■ No other physical interface has the same subnet as the specified interfaces. If the value of the attribute is 1, the MultiNICB agent also assigns all the specified interfaces in one multi-pathing group. Valid values for this attribute are 0 and 1. Type and dimension: integer-scalar Default: 1

Table 3-14 Optional attributes for Multi-Pathing mode

Optional attribute	Description
MpathdRestart	If the value of the attribute is 1, MultiNICB tries to restart mpathd. Valid values for this attribute are 0 and 1. Type and dimension: integer-scalar Default: 1

Resource type definition

```
type MultiNICB (
    static int MonitorInterval = 10
    static int OfflineMonitorInterval = 60
    static str Operations = None
    static str ArgList[] = { UseMpathd, MpathdCommand, ConfigCheck,
    MpathdRestart, Device, NetworkHosts, LinkTestRatio,
    IgnoreLinkStatus, NetworkTimeout, OnlineTestRepeatCount,
    OfflineTestRepeatCount, NoBroadcast, DefaultRouter, Failback,
    GroupName, Protocol }
    int UseMpathd
    str MpathdCommand = "/usr/lib/inet/in.mpathd"
    int ConfigCheck = 1
    int MpathdRestart = 1
    str Device{}
    str NetworkHosts[]
    int LinkTestRatio = 1
    int IgnoreLinkStatus = 1
    int NetworkTimeout = 100
    int OnlineTestRepeatCount = 3
    int OfflineTestRepeatCount = 3
    int NoBroadcast
    str DefaultRouter = "0.0.0.0"
    int Failback
    str GroupName
    str Protocol = IPv4
)
```

Solaris operating modes: Base and Multi-Pathing

The MultiNICB agent has two modes of operation, Base and Multi-Pathing, which you can set with the UseMpathd attribute.

Base mode

The value of the UseMpathd attribute is 0 by default for this mode. In Base mode, to monitor the interfaces that it controls, the agent:

- sends the packets to other hosts on the network for probe-based detection
- tests the link status of the interfaces for link-based detection
- checks that all the specified interfaces are in the same IP subnet and group, and have "NOFAILOVER" and "DEPRECATED" flags set on them
- checks that no other physical interface has the same subnet as the specified interfaces
- assigns all the specified interfaces in one multi-pathing group

The agent logs link failures and failovers when it uses either link- or probe-based detection.

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 if the Failback attribute for the corresponding MultiNICB resource is set to 1.

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

The MultiNICB agent, in Base mode, uses the following criteria to determine if an interface works:

- Link-based detection of the interface status
  The interface driver reports the status of the link. Note that not all drivers support this feature. Set the value of IgnoreLinkStatus to 1 to disable this test.
- Probe-based detection using Internet Control Message Protocol (ICMP) echo
  Set the LinkTestRatio attribute to a value greater than 0 to send ICMP echo request packets to a specified network host. You specify the network hosts in the NetworkHosts attribute. You must assign test IP addresses to the interface for probe-based detection. The test IP address is needed to send the ICMP packets, which determines the link's status. If you set the value of the LinkTestRatio attribute to 0, you do not need to assign test IP addresses.
  If you specify no hosts in the NetworkHosts attribute, the agent uses the ICMP broadcast when the value of the NoBroadcast attribute is 0. It caches the sender of the first reply for future use as a network host. While the agent sends and receives ICMP packets, the IP layer is completely bypassed.

You can assign addresses and still do only link-based detection by setting the values of the LinkTestRatio attribute to greater than 0 and the IgnoreLinkStatus attribute to 0. You can skip link-based detection (link driver tests) and only do ICMP tests if:

- the value of the IgnoreLinkStatus attribute is 1, and
- the value of the LimitTestRation attribute is greater than 0, and
- the test IP addresses are assigned to the interface

The MultiNICB agent performs both link-based detection and probe-based detection if:

- the value of the LinkTestRatio attribute is greater than 0, and
- the value of the IgnoreLinkStatus attribute is 0, and
- the test IP addresses are assigned to the interface

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

Failover and failback

During an interface failure, the MultiNICB agent fails over all logical IP addresses to a working interface under the same resource. The agent remembers the first physical interface from which an IP address was failed over. This physical interface becomes the “original” interface for the particular logical IP address. When the original interface is repaired, the logical IP address fails back to it if the Failback attribute is set to 1.

Multi-Pathing mode

To activate this mode set the value of the UseMpathd attribute to 1. The MultiNICB agent, in Multi-Pathing mode, monitors Oracle’s IP Multi-Pathing daemon (mpathd). The MultiNICB agent specifically monitors the FAILED flag on physical interfaces and the mpathd process. See the man page: in.mpathd (1M) for more information on this daemon.

Oracle’s mpathd daemon monitors the interfaces that are part of the IPMP group. The daemon:

- sends the packets to other hosts on the network for probe-based detection as long as a test IP address is assigned to the network interface
- checks the link status of the interfaces for link-based detection as long as the interface supports the test for detection

The mpathd daemon can perform both link- and probe-based detection when test IP addresses are assigned to NIC interfaces.

The MultiNICB agent logs errors when the daemon is not running, or if a configuration path error exits. The mpathd daemon logs link failures and IP address failovers in the system log.

Trigger script

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

- MultiNICB resource name
- The device whose status changed, for example:
  - qfe0
- 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: Interface qfe0 came up.
- The script saves last IP address-to-interface name association. If any of the IP addresses have been moved, added, or removed, it prints out a message to the console, for example: MultiNICB: IP address 192.4.3.3 moved from interface qfe1:1 to interface qfe0:1

Sample configurations

Interface configuration for Solaris

Set the EPROM variable to assign unique MAC addresses to all ethernet interfaces on the host:

```
# eeprom local-mac-address?=true
```

Reboot the system after setting the eprom variable to complete the address setup. The base IP addresses must be configured on the interfaces before the MultiNICB agent controls the interfaces. You can configure these addresses at system start up using /etc/hostname.*XXX* initialization files. Refer to the following examples for more information.

Setting up test IP addresses for Base Mode

These examples demonstrate setting up test IP addresses for your clustered systems. These IP addresses allow the agent determine if the NIC works. The agent determines that the NIC works if it receives responses for the ping packets that it sends to other nodes on the network. You do *not* need to perform the following steps for the floating IP addresses. The agent performs these steps.

In the file /etc/hostname.qfe0, add the following two lines:

```
north-qfe0 netmask + broadcast + deprecated -failover up \
        addif north netmask + broadcast + up
```

Where north-qfe0 is the test IP address that the agent uses to determine the state of the qfe0 network card.

In the file /etc/hostname.qfe4, add the following line:

```
north-qfe4 netmask + broadcast + deprecated -failover up
```

Where north-qfe4 is the test IP address that the agent uses to determine the state of the qfe4 network card.

In the example, north-qfe0 and north-qfe4 are the host names that correspond to test IP addresses. north is the host name that corresponds to the test IP address.

IPMultiNICB and MultiNICB configuration

```
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 g11_i1 (
    BaseResName = gnic_n
    Address = "192.1.0.201"
    NetMask = "255.255.0.0"
    DeviceChoice = "1"
)
Proxy g11_p1 (
    TargetResName = gnic_n
)
g11_i1 requires g11_p1

// A parallel group for the MultiNICB resource

group gnic (
    SystemList = { north = 0, south = 1 }
    AutoStartList = { north, south }
    Parallel = 1
)
MultiNICB gnic_n (
    Device @north = { qfe0 = 0, qfe4 = 1 }
    Device @south = { qfe0 = 0, qfe4 = 1 }
    NetworkHosts = { "192.1.0.1" }
)
Phantom gnic_p (
)
```

IPv6 configuration

The following is a basic configuration for IPv6 with IPMultiNICB and MultiNICB resources.

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

        IPMultiNICB ip_res (
            BaseResName = nic_res
            Address = "3ffe:21::90:1211:161"
            PrefixLen = 64
            )

        MultiNICB nic_res (
            Device = { e1000g1 = 0, e1000g3 = 1 }
            Failback = 1
            Protocol = IPV6
            )
```

Debug log levels

The MultiNICB agent uses the following debug log levels:

DBG_1, DBG_2, DBG_3, DBG_4, DBG_5

DNS agent

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

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

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

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

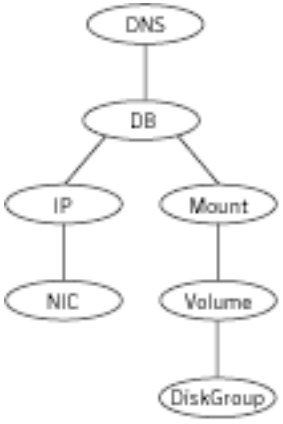
For important information about this agent, refer to:

"DNS agent notes" on page 141

Dependencies

No dependencies exist for the DNS resource.

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



Agent functions

Online	Updates one or more name servers with the resource records. The agent updates the name servers defined in the StealthMasters attribute. If you have not configured this attribute then the agent obtains the name of the master server by sending an Start of Authority (SOA) query. This query retrieves the SOA record of the zone defined in the agent's Domain attribute. This SOA record contains the name of the master server. The agent creates PTR records for each RR of type A or AAAA if the value of the CreatePTR attribute is true. A prerequisite for this feature is that the same master or stealth server serves the forward (A or AAAA) and reverse zones. Finally the agent generates an Online lock file to indicate that the resource is online on the current system.
Offline	Removes the Online lock file. If attribute OffDelRR is true, offline removes all records that the ResRecord keys define.
Monitor	Returns the ONLINE state if at least one name server reports all mappings that ResRecord defines. The name servers are the master or StealthMaster servers and all the servers for which an NS record for the zone exists.
Clean	Removes the Online lock file, if it exists.
Open	Removes the Online lock file if the resource is reported online on another node inside the cluster to prevent concurrency violation. If the lock file exists, at least one name server has to report all the records that the ResRecord attribute defines. If all the name servers fail to report all the records, the agent function removes the Online lock file.
Action	Different action agent functions follow: ■ keyfile.vfd This action entry point checks if the key file as specified in the TSIGKeyFile attribute exists either locally or on shared storage. ■ dig.vfd This action entry point checks if dig and nsupdate binaries exist and are executable. ■ master.vfd This action entry point checks if stealth masters are able to reply to SOA query for the configured domain.

State definitions

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

Attributes

Table 3-15 Required attributes

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

Table 3-15 Required attributes

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

Table 3-16 Required attributes

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

Table 3-17 Optional attributes

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

Table 3-17 Optional attributes

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

Table 3-17 Optional attributes

Optional attribute	Description
OffDelRR	Use the OffDelRR attribute to direct the offline agent function to remove all records that the ResRecord key defines. You must set the value of this attribute to true (1) to have the agent remove all the records. The online agent function always adds records if they do not exist. Type and dimension: boolean-scalar Default: 0 Example: 1

Resource type definition

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

DNS agent notes

The DNS agent has the following notes:

- "High availability fire drill" on page 142
- "Monitor scenarios" on page 142
- "Sample Web server configuration" on page 142
- "Secure DNS update for BIND 9" on page 143
- "Setting up secure updates using TSIG keys for BIND 9" on page 143

High availability fire drill

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

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

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

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

Monitor scenarios

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

Table 3-18 Monitor scenarios for the Online lock file

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

Sample Web server configuration

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

Secure DNS update for BIND 9

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

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

Setting up secure updates using TSIG keys for BIND 9

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

To use secure updates using TSIG keys

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

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

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

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

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

   ```
   +Cdjlkef9ZTSeixERZ433Q==
   ```

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

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

   Where `+Cdjlkef9ZTSeixERZ433Q==` is the key.

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

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

6. Save and restart the named process.

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

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

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

Sample configurations

This section contains sample configurations for this agent.

Basic IPv6 configuration

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

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

    DNS ipv6group_dns_res (
        Critical = 0
        Domain = "ipv6.vcs.net"
        TSIGKeyFile =
        "/var/tsig/Kipv6.vcscd.net.+157+18435.private"
        StealthMasters = { "2001:db8:c18:2:69c4:3251:bac1:6cbe" }
        ResRecord = {
            vcssysCv6 = "2001:db8:c18:2:214:4fff:fe96:8833",
            sysC = vcssysCv6 }
        )

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

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

    ipv6group_dns_res requires ipv6group_ip_res
    ipv6group_ip_res requires ipv6group_nic_res
```

IPv6 CNAME sample configuration

The following sample configuration uses CNAME values.

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

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

IPv4 A sample configuration

The following sample configuration uses A values.

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

    DNS forward_group_v4_resource (
        Domain = "example.com"
        StealthMasters = { "3ffe:556::1000:5761" }
        ResRecord @sysA = { www = "10.200.56.240" }
        ResRecord @sysB = { www = "10.200.56.244" }
        OffDelRR = 1
        )
```

Debug log levels

The DNS agent uses the following debug log levels:

DBG_1, DBG_2, DBG_3, DBG_4, DBG_5

Chapter 4

File share agents

This chapter contains the following:

- "About the file service agents" on page 147
- "NFS agent" on page 148
- "NFSRestart agent" on page 153
- "Share agent" on page 162
- "About the Samba agents" on page 166
- "SambaServer agent" on page 168
- "SambaShare agent" on page 173
- "NetBios agent" on page 176

About the file service agents

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

NFS agent

Starts and monitors the nfsd and mountd 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.

Note: VCS supports a maximum of one service group with NFS shares when the UseSMF attribute's value is 1.

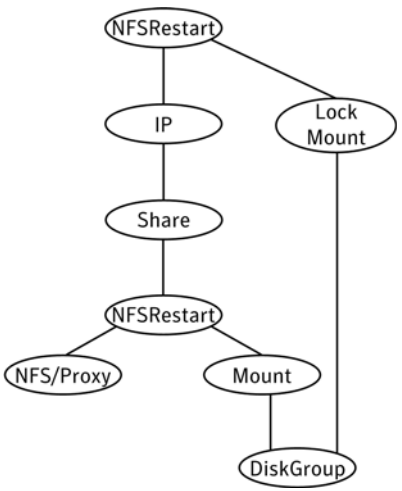
For important information about this agent, refer to:

"NFS agent notes" on page 151

Dependencies

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

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



Agent functions

Online	Checks if nfsd, mountd, and nfsmapid (nfsmapid is for Solaris 10) daemons are running. If they are not running, the agent starts the daemons.
Monitor	Monitors versions 2, 3, and 4 of the nfsd daemons, and versions 1, 2, and 3 of the mountd daemons. Monitors TCP and UDP versions of the daemons by sending RPC (Remote Procedure Call) calls `clnt_create` and `clnt_call` to the RPC server. If the calls succeed, the resource is reported ONLINE.
Clean	Terminates and restarts the nfsd, mountd, and nfsmapid daemons.

State definitions

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

Attributes

Optional attributes

Optional attributes	Description
CleanRmtab	To clear the /etc/rmtab file before staring the mount daemon, set the value of CleanRmtab to 1. Type and dimension: boolean-scalar Default: 0

Optional attributes

Optional attributes	Description
LockFileTimeout	Specifies the time period in seconds after which the agent deletes the lock files. The agent maintains the files internally to synchronize the starting and stopping of NFS daemons between multiple service groups. Set this value to the total time needed for a service group to go offline or come online on a node. In situations where you have multiple service groups, set this value for the service group that takes the longest time. Type and dimension: integer-scalar Default: 180 Example: 240
Nservers	Specifies the number of concurrent NFS requests the server can handle. Type and dimension: integer-scalar Default: 16 Example: 24
UseSMF	For Solaris 10, the value of this attribute determines whether you want to enable the resource to use Service Management Facility (SMF) for NFS daemons. Set the value of the UseSMF attribute to 1, to enable the resource to use SMF. Accept the default value of 0 if you do not want to enable the resource to use SMF. You must keep the NFS daemons under VCS control however. **Note:** When you use a SMF service to enable the NFS server, a service issue requires that at least one entry is in /etc/dfs/dfstab. See "Using Service Management Facility (SMF) to control NFS daemons for Solaris 10" on page 151. Type and dimension: boolean-scalar Default: 0

Resource type definition

```
type NFS (
    static int RestartLimit = 1
    static str ArgList[] = { UseSMF, Nservers, LockFileTimeout,
    CleanRmtab }
    static str Operations = OnOnly
    int Nservers = 16
    int LockFileTimeout = 180
    boolean UseSMF = 0
    boolean CleanRmtab = 0
)
```

NFS agent notes

The NFS agent has the following notes:

- "Using Service Management Facility (SMF) to control NFS daemons for Solaris 10" on page 151

Using Service Management Facility (SMF) to control NFS daemons for Solaris 10

On Solaris 10, if you do not want to use SMF for NFS daemon control, disable SMF for these daemons.

SMF use for NFS resources is disabled by default. If you plan to use it for NFS resources, you must set the value of the UseSMF attribute to a value of 1. If you set the value of the UseSMF attribute to 1, you do not have to perform the following instructions. If you keep the default value of 0 for the UseSMF attribute, you must perform the following instructions.

UNIX start-up scripts and configuration files previously performed these functions. SMF maintains the Service Configuration Repository to store persistent configuration information as well as runtime data for all the services. All NFS daemons (nfsd, mountd, etc.) are now controlled by SMF. To keep these daemons under VCS control, modify the configuration repository to disable the SMF framework for NFS daemons.

You must invoke the following command before bringing the NFS agent online or the agents returns an UNKNOWN state.

To keep nfsd and mountd daemons under VCS control

1. Set the auto_enable property to false.

   ```
   # svccfg -s nfs/server setprop "application/auto_enable = false"
   ```

2. Refresh the SMF configuration repository.

   ```
   # svcadm refresh nfs/server
   ```

3. Disable SMF.

```
# svcadm disable svc:/network/nfs/server:default
```

4 Run `svcs -a | grep -i nfs` command and review its output to make sure that SMF for nfsd and mountd is disabled.

```
# svcs -a | grep -i nfs
disabled       May_29   svc:/network/nfs/server:default
```

To keep nfsmapid daemon under VCS control

1 Set the auto_enable property to false.

```
# svccfg -s nfs/mapid setprop "application/auto_enable = false"
```

2 Refresh the SMF configuration repository.

```
# svcadm refresh nfs/mapid
```

3 Disable SMF.

```
# svcadm disable svc:/network/nfs/mapid:default
```

4 Run `svcs -a | grep -i mapid` command and review its output to make sure that SMF for nfsmapid is disabled.

```
# svcs -a | grep -i mapid
disabled       May_29   svc:/network/nfs/mapid:default
```

Sample configurations

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

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

Debug log levels

The NFS agent uses the following debug log levels:

DBG_1, DBG_3, DBG_4, DBG_5

NFSRestart agent

The NFSRestart agent provides the following functionalities:

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

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

For important information about this agent, refer to:

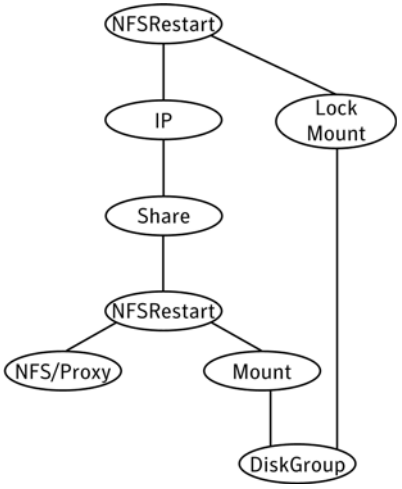
"NFSRestart agent notes" on page 157

Dependencies

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

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

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



Agent functions

Online

For the lower NFSRestart resource:

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

For the upper NFSRestart resource:

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

Monitor

For the lower NFSRestart resource:

The monitor agent function does nothing.

For the upper NFSRestart resource:

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

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

State definitions

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

Attributes

Table 4-1 Required attributes

Required attribute	Description
NFSRes	Name of the NFS resource. Do not set this to the name of the Proxy resource that points to the NFS resource. Type and dimension: string-scalar Example: "nfsres1"

Table 4-2 Optional attributes

Required attribute	Description
LocksPathName	The path name of the directory to store the NFS locks for all the shared file systems. You can use the pathname of one of the shared file systems for this value. 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
LockServers	The maximum number of concurrent lockd threads for execution on a server. Type and dimension: integer-scalar Default: "20" Example: "25"

Table 4-2 Optional attributes

Required attribute	Description
Lower	Defines the position of NFSRestart resource in the service group. The NFSRestart resource below the Share resource needs a value of 1. The NFSRestart resource on the top of the resource dependency tree has a Lower attribute value of 0. Type and dimension: integer-scalar Default: 0

Resource type definition

```
type NFSRestart (
    static str ArgList[] = { LocksPathName, NFSLockFailover,
    LockServers, NFSRes, "NFSRes:Nservers",
    "NFSRes:LockFileTimeout", "NFSRes:UseSMF", Lower, State }
    static keylist SupportedActions = { "lockdir.vfd", "nfsconf.vfd"
    }
    str NFSRes
    str LocksPathName
    boolean NFSLockFailover = 0
    int LockServers = 20
    int Lower = 0
)
```

NFSRestart agent notes

The NFSRestart agent has the following notes:

- "About high availability fire drill" on page 157
- "Mounting NFS export with the -vers options can cause lock failure" on page 158
- "Providing a fully qualified host name" on page 158
- "Service Management Facility—Solaris 10" on page 159

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*.

Mounting NFS export with the -vers options can cause lock failure

For Solaris clients, the NFS lock recovery may fail to work if the NFS export is mounted with the "-vers=3" option.

Providing a fully qualified host name

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

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

Service Management Facility—Solaris 10

You need to enable the NFS attribute UseSMF to enable monitoring of NFS and Lock daemons through Service Management Facility in Solaris.

If the UseSMF NFS attribute is not enabled, then you must disable the Service Management Facility (SMF) for NFS daemons for the NFSRestart agent to work on Solaris 10. SMF is the service framework for Solaris 10 starting from build 64. SMF provides an infrastructure to automatically start and restart services. Previously, UNIX start-up scripts and configuration files performed these functions.

SMF maintains the Service Configuration Repository, which stores persistent configuration information and runtime data for all the services. Thus, SMF now controls all NFS locking daemons (lockd, statd, etc.) To keep these daemons under VCS control, you need to modify the configuration repository to disable the SMF framework for NFS daemons.

You must invoke the following command before bringing the NFSRestart agent online or the agents returns an UNKNOWN state.

To keep the statd daemon under VCS control

1. Set the auto_enable property to false.

   ```
   # svccfg -s nfs/status setprop "application/auto_enable = false"
   ```

2. Refresh the SMF configuration repository.

   ```
   # svcadm refresh nfs/status
   ```

3. Disable SMF.

   ```
   # svcadm disable svc:/network/nfs/status:default
   ```

4. Run `svcs -a | grep -i nfs` command and review its output to make sure that SMF for statd is disabled.

   ```
   # svcs -a | grep -i nfs
   disabled       May_29   svc:/network/nfs/status:default
   ```

To keep lockd daemon under VCS control

1. Set the auto_enable property to false.

```
# svccfg -s nfs/nlockmgr setprop "application/auto_enable =
false"
```

2 Refresh the SMF configuration repository.

```
# svcadm refresh nfs/nlockmgr
```

3 Disable SMF.

```
# svcadm disable svc:/network/nfs/nlockmgr:default
```

4 Run `svcs -a | grep -i nfs` command and review its output to make sure that SMF for nlockmgr is disabled.

```
# svcs -a | grep -i nlockmgr
disabled       May_29   svc:/network/nfs/nlockmgr:default
```

To manually restart lockd, statd, and automountd

- For lockd:

```
# /usr/lib/nfs/lockd
```

- For statd:

```
# /usr/lib/nfs/statd
```

- For automountd:

```
# /usr/lib/fs/autofs/automount
# /usr/lib/autofs/automountd
```

Sample configurations

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

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

Basic agent configurations

For NFS lock recovery:

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

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

For no NFS lock recovery:

```
NFSRestart nfsrestart (
    NFSRes = nfsres
    )

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

Debug log levels

The NFSRestart agent uses the following debug log levels:

DBG_1, DBG_3, DBG_4, DBG_5

Share agent

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

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

For important information on this agent, refer to:

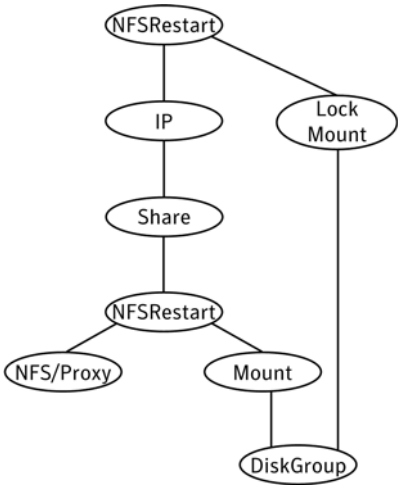
"Share agent notes" on page 164

Dependencies

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

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

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



Agent functions

Online	Shares an NFS file system.
Offline	Unshares an NFS file system.

Monitor	Reads /etc/dfs/sharetab file and looks for an entry for the file system specified by PathName. If the entry exists, monitor returns ONLINE.
Clean	Terminates all ongoing resource actions and takes the resource offline, forcibly when necessary.
Action	■ direxists.vfd Checks if the path specified by the PathName attribute exists on the cluster node. If the path name is not specified, it checks if a corresponding mount point is available to ensure that the path is on shared storage.

State definitions

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

Attributes

Table 4-3 Required attributes

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

Table 4-4 Optional attributes

Optional attribute	Description
Options	Options for the `share` command. Type and dimension: string-scalar Example: "-o rw"

Resource type definition

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

Share agent notes

The following section contains notes on the Share agent.

High availability fire drill

The high availability fire drill detects discrepancies between the VCS configuration and the underlying infrastructure on a node. These discrepancies

might prevent a service group from going online on a specific node. For Share resources, the high availability fire drill checks if the path exists.

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

Sample configurations

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

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

Debug log levels

The Share agent uses the following debug log levels:

DBG_1, DBG_3, DBG_5

About the Samba agents

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

- Filespace
- Printer
- WINS
- Domain Master

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

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

The Samba agents

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

Before using the Samba agents

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

Supported versions

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

```
# smbd -V
```

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

Notes for configuring the Samba agents

The following notes describe configuration considerations for the Samba agents.

Configuring multiple SambaServer resources

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

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

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

To check for the default value of the Samba configuration file

- Enter the following command:

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

To check for the default location of the Samba pidfile

- Enter the following command:

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

SambaServer agent

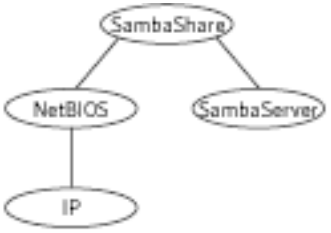
The SambaServer agent starts, stops, and monitors the smbd process as a daemon. You can use the agent to make a smbd daemon highly available.

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

Dependencies

No dependencies exist for the SambaServer resource.

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



Agent functions

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

State definitions

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

Attributes

Table 4-5 Required attributes

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

Table 4-6 Optional attributes

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

Table 4-6 Optional attributes

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

Resource type definitions

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

Sample configurations

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

Debug log levels

The SambaServer agent uses the following debug log levels:

DBG_1, DBG_2, DBG_3, DBG_4, DBG_5

SambaShare agent

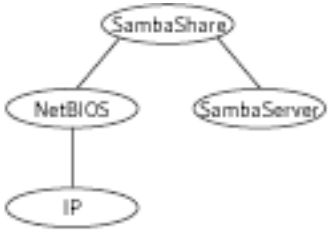
The SambaShare agent adds, removes, and monitors a share by modifying the specified Samba configuration file. You can use the agent to make a Samba Share highly available.

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

Dependencies

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

Figure 4-5 Sample service group for a SambaShare resource



Agent functions

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

State definitions

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

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

Attributes

Table 4-7 Required attributes

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

Resource type definition

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

Sample configuration

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

Debug log levels

The SambaShare agent uses the following debug log levels:

DBG_1, DBG_5

NetBios agent

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

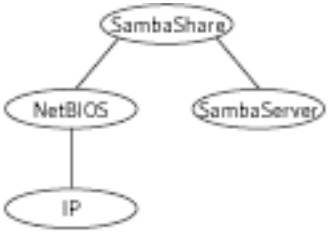
The agent sets, monitors, and resets the names and network interfaces by which the Samba server is known. The agent also sets, monitors and resets Samba to act as a WINS server or domain master or both.

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

Dependencies

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

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



Agent functions

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

State definitions

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

OFFLINE — Indicates one or more of the following:

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

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

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

Attributes

Table 4-8 Required attributes

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

Table 4-9 Optional attributes

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

Table 4-9 Optional attributes

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

Resource type definition

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

Sample configuration

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

Debug log levels

The NetBios agent uses the following debug log levels:

DBG_1, DBG_5

Chapter 5

Service and application agents

This chapter contains the following agents:

- "About the service and application agents" on page 181
- "Apache Web server agent" on page 182
- "Application agent" on page 194
- "CoordPoint agent" on page 204
- "Process agent" on page 208
- "ProcessOnOnly agent" on page 213
- "Zone agent" on page 216
- "LDom agent" on page 221
- "Project agent" on page 226

About the service and application agents

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

Apache Web server agent

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

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

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

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

The ContainerName and ContainerType attributes are deprecated. For more information on using zones, refer to the *Veritas Storage Foundation and High Availability Solutions Virtualization Guide*.

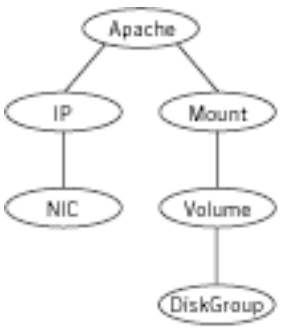
For more information regarding this agent:

See "Apache Web server notes" on page 189.

Dependencies

This type of resource depends on IP and Mount resources.

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



Agent functions

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

State definitions

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

Attributes

Table 5-1 Required attributes

Required attribute	Description
ConfigFile	Full path and file name of the main configuration file for the Apache server. Type and dimension: string-scalar Example: "/apache/server1/conf/httpd.conf"
httpdDir	Full path of the directory to the httpd binary file Type and dimension: string-scalar Example: "/apache/server1/bin"
ResLogLevel	Controls the agent's logging detail for a specific instance of a resource. Values are: ■ ERROR: Logs error messages. ■ WARN: Logs error and warning messages. ■ INFO: Logs error, warning, and informational messages. ■ TRACE: Logs error, warning, informational, and trace messages. Trace logging is verbose. Use for initial configuration or troubleshooting. Type and dimension: string-scalar Default: INFO Example: "TRACE"
PidFile	This attribute is required when you want to enable the detection of a graceful shutdown outside of VCS control. See "PidFile" on page 187.
EnvFile	This attribute may be required when you use IBM HTTP Server. See "EnvFile" on page 186.

Table 5-2 Optional attributes

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

Table 5-2 Optional attributes

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

Table 5-2 Optional attributes

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

Table 5-2 Optional attributes

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

Table 5-3 Resource type attribute

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

Resource type definition

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

```
    str DirectiveBefore{}
    static int ContainerOpts{} = { RunInContainer=1, PassCInfo=0 }
    static boolean IntentionalOffline = 0
)
```

Apache Web server notes

The Apache Web server has the following notes:

- "Tasks to perform before you use the Apache Web server agent" on page 189
- "About detecting application failure" on page 190
- "About bringing an Apache Web server online outside of VCS control" on page 190
- "About high Availability fire drill" on page 190

Tasks to perform before you use the Apache Web server agent

Before you use this agent, perform the following tasks:

- Install the Apache server on shared or local disks.
- Ensure that you are able to start the Apache Web server outside of VCS control, with the specified parameters in the Apache configuration file (for example: /etc/apache/httpd.conf). For more information on how to start the server:
  See "About bringing an Apache Web server online outside of VCS control" on page 190.
- Specify the location of the error log file in the Apache configuration file for your convenience (for example: ErrorLog /var/apache/logs/error_log).
- Verify that the floating IP has the same subnet as the cluster systems.
- If you use a port other than the default 80, assign an exclusive port for the Apache server.
- Verify that the Apache server configuration files are identical on all cluster systems.
- Verify that the Apache server does not autostart on system startup.
- Verify that `Inetd` does not invoke the Apache server.
- Remove previous versions of this agent.
- The service group has disk and network resources to support the Apache server resource.

- Assign virtual host name and port to Apache Server.

About detecting application failure

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

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

About bringing an Apache Web server online outside of VCS control

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

To start an Apache Web server outside of VCS control

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

About high Availability fire drill

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

resources, when the Apache Web server is installed locally, the high availability fire drill checks for the validity of these attributes:

- ConfigFile
- httpdDir

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

Sample configurations

Basic configuration for Solaris

The following is a basic configuration for the resource.

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

        Apache httpd_server (
                Critical = 0
                httpdDir = "/apache/bin"
                HostName = vcsso11
                Port = 8888
                User = root
                SecondLevelMonitor = 1
                ConfigFile = "/apache/conf/httpd.conf"
                )

        DiskGroup Apache_dg (
                Critical = 0
                DiskGroup = apc1
                )

        IP Apache_ip (
                Critical = 0
                Device = bge0
                Address = "11.123.99.168"
                NetMask = "255.255.254.0"
                )

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

        Apache_mnt requires Apache_dg
        httpd_server requires Apache_mnt
```

```
httpd_server requires Apache_ip
```

Basic IPv6 configuration

The following is a basic IPv6 configuration for the resource.

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

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

    DiskGroup ipv6group_dg_res (
        DiskGroup = dg01
        )

    IP ipv6group_ip_res (
        Device = bge0
        Address = "fd4b:454e:205a:110:211:25ff:fe7e:118"
        PrefixLen = 64
        )

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

    NIC ipv6group_nic_res (
        Device = bge0
        )

    Volume ipv6group_vol_res (
        Volume = vol01
        DiskGroup = dg01
        )

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

Application agent

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

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

You can monitor the application in the following ways:

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

The ContainerName and ContainerType attributes are deprecated.

This agent is zone-aware. The ContainerOpts resource type attribute for this type has a default value of 1 for RunInContainer and a default value of 0 for PassCInfo. Symantec recommends that you do not change these values. Refer to the *Storage Foundation High Availability Virtualization Guide* for more information.

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

High availability fire drill

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

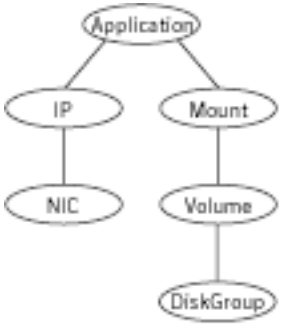
- The availability of the specified program (program.vfd)
- Execution permissions for the specified program (program.vfd)
- The existence of the specified user on the host (user.vfd)
- The existence of the same binary on all nodes (cksum.vfd)

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

Dependencies

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

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



Agent functions

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

To bring the resource online, the agent function performs the command:

```
su [-] user -c command_to_online_resource
```

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

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

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

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

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

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

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

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

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

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

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

Clean — Terminates processes specified in PidFiles or MonitorProcesses. Ensures that only those processes (that are specified in the MonitorProcesses attribute) running with the user ID specified in the User attribute are killed. If the CleanProgram is defined, the agent executes the CleanProgram. Note that if none of the PidFiles, MonitorProcesses, or CleanProgram are specified for the resource, the Application agent uses StopProgram during the clean operation.

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

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

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

State definitions

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

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

UNKNOWN — Indicates an indeterminable application state or invalid configuration or that the required attributes have not been configured.

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

Attributes

Table 5-4 Required attributes

Required attribute	Description
StartProgram	The executable, created locally on each node, which starts the application. Specify the complete path of the executable. Applicable command line arguments follow the name of the executable and have spaces separating them. For applications running in Solaris 10 zones, use the path as seen from the non-global zone. **Note:** Do not use the opening and closing ({ }) brace symbols in this string. Type and dimension: string-scalar Example: "/usr/sbin/sample_app start"
StopProgram	The executable, created locally on each node, which stops the application. Specify the complete path of the executable. Applicable command line arguments follow the name of the executable and have spaces separating them. For applications running in Solaris 10 zones, use the path as seen from the non-global zone. **Note:** Do not use the opening and closing ({ }) brace symbols in this string. Type and dimension: string-scalar Example: "/usr/sbin/sample_app stop"
At least one of the following attributes: ■ MonitorProcesses ■ MonitorProgram ■ PidFiles	See “Optional attributes” on page 199.

Table 5-5 Optional attributes

Optional attribute	Description
CleanProgram	The executable, created locally on each node, which forcibly stops the application. Specify the complete path of the executable. Applicable command line arguments follow the name of the executable and have spaces separating them. For applications running in Solaris 10 zones, use the path as seen from the non-global zone. Type and dimension: string-scalar Example: "/usr/sbin/sample_app force stop"
MonitorProcesses	A list of processes that you want monitored and cleaned. Each process name is the name of an executable. Provide the full path name of the executable if the agent uses that path to start the executable. The process name must be the full command line argument that the `ps -u user -o args` command displays for the process. Type and dimension: string-vector Example: "nmbd"
MonitorProgram	The executable, created locally on each node, which monitors the application. Specify the complete path of the executable. Applicable command line arguments follow the name of the executable and have spaces separating them. For applications running in Solaris 10 zones, use the path as seen from the non-global zone. MonitorProgram can return the following VCSAgResState values: OFFLINE value is 100; ONLINE values range from 101 to 110 (depending on the confidence level); 110 equals confidence level of 100%. Any other value = UNKNOWN. **Note:** Do not use the opening and closing ({ }) brace symbols in this string. Type and dimension: string-scalar Example: "/usr/sbin/sample_app_monitor all"

Table 5-5 Optional attributes

Optional attribute	Description
PidFiles	A list of files that contain the PID (process ID) of the processes that you want monitored and cleaned. These are application generated files. Each PID file contains one monitored PID. Specify the complete path of each PID file in the list. For applications running in Solaris 10 non-global zones, include the zone root path in the PID file's path—the global zone's absolute path. The process ID can change when the process restarts. If the application takes time to update the PID file, the agent's monitor function may return an incorrect result. If incorrect results occur, increase the ToleranceLimit in the resource definition. Type and dimension: string-vector Example: "/var/lock/samba/smbd.pid" Example in a global zone for Solaris 10: "/var/lock/samba/smbd.pid" Example in a non-global zone for Solaris 10: "$*zoneroot*/var/lock/samba/smbd.pid" Where the `$zoneroot` is the root directory of the non-global zone, as seen from the global zone.
User	The user name for running StartProgram, StopProgram, MonitorProgram, and CleanProgram. The processes that are specified in the MonitorProcesses list must run in the context of the specified user. Monitor checks the processes to make sure they run in this context. Type and dimension: string-scalar Default: root Example: user1

Table 5-5 Optional attributes

Optional attribute	Description
EnvFile	The environment file that should get sourced before running any of the StartProgram, StopProgram, MonitorProgram or CleanProgram. Type and dimension: string-scalar Default: "" Example: /home/username/envfile
UseSUDash	When the value of this attribute is 0, the agent performs an `su User` command before it executes the StartProgram, the StopProgram, the MonitorProgram, or the CleanProgram agent functions. When the value of this attribute is 1, the agent performs an `su - User` command before it executes the StartProgram, the StopProgram, the MonitorProgram or the CleanProgram agent functions. Type and dimension: boolean-scalar Default: 0 Example: 1

Resource type definition

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

```
    boolean UseSUDash = 0
)
```

Application agent notes

Using Application agent with IMF

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

Table 5-6 Relation of monitoring attributes with IMF modes

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

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

Sample configurations

Configuration 1

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

```
Application samba_app (
```

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

Configuration 2

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

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

Configuration 3 for Solaris 10

In this example, configure a resource in a non-global zone: zone1. The ZonePath of zone1 is /zone1/root. Configure the executable samba as StartProgram and StopProgram, with start and stop specified as command line arguments respectively. Configure the agent to monitor two processes: a process that the smbd.pid specifies and the process nmbd.

```
Application samba_app (
    StartProgram = "/usr/sbin/samba start"
    StopProgram = "/usr/sbin/samba stop"
    PidFiles = { "/zone1/root/var/lock/samba/smbd.pid" }
    MonitorProcesses = { "nmbd" }
)
```

Debug log levels

The Application agent uses the following debug log levels:

DBG_1, DBG_2, DBG_3, DBG_4, DBG_5

CoordPoint agent

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

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

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

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

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

For important information about this agent, refer to:

"Notes for the CoordPoint agent" on page 206

Dependencies

No dependencies exist for the CoordPoint resource.

Agent functions

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

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

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

See "CoordPoint agent I/O fencing reporting activities" on page 206.

State definitions

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

Attributes

Table 5-7 Required attributes

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

Resource type definition

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

Notes for the CoordPoint agent

CoordPoint agent I/O fencing reporting activities

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

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

AutoStartList attribute

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

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

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

Sample configuration

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

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

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

Debug log levels

The CoordPoint agent uses the following debug log levels:

DBG_1, DBG_10

Process agent

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

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

The ContainerName and ContainerType attributes are deprecated.

This agent is zone-aware. The ContainerOpts resource type attribute for this type has a default value of 1 for RunInContainer and a default value of 0 for PassCInfo. Symantec recommends that you do not change these values. Refer to the *Storage Foundation High Availability Virtualization Guide* for more information.

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

High availability fire drill

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

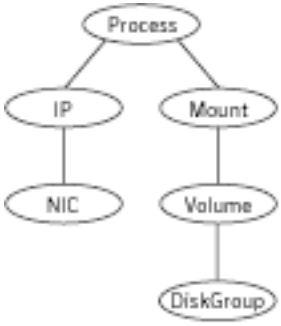
- The existence of a binary executable for the specified process (program.vfd)
- The existence of the same binary on all nodes (program.vfd)

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

Dependencies

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

Figure 5-3 Sample service group for a Process resource



Agent functions

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

State definitions

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

Attributes

Table 5-8 Required attribute

Required attribute	Description
PathName	Complete pathname to access an executable program. This path includes the program name. If a script controls the process, the PathName defines the complete path to the shell. This attribute must not exceed 80 characters. Type and dimension: string-scalar Example: "/usr/lib/sendmail"

Table 5-9 Optional attributes

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

Resource type definition

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

Sample configurations

Configuration 1

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

Configuration 2

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

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

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

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

Debug log levels

The Process agent uses the following debug log levels:

DBG_1, DBG_4, DBG_5

ProcessOnOnly agent

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

The ContainerName and ContainerType attributes are deprecated.

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

Dependencies

No child dependencies exist for this resource.

Agent functions

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

State definitions

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

Attributes

Table 5-10 Required attributes

Required attribute	Description
PathName	Defines complete pathname to access an executable program. This path includes the program name. If a process is controlled by a script, the PathName defines the complete path to the shell. Pathname must not exceed 80 characters. The value configured for this attribute needs to match the process listing from the ps output for the agent to display as ONLINE. Type and dimension: string-scalar Example: "/usr/lib/nfs/nfsd"

Table 5-11 Optional attributes

Optional attribute	Description
Arguments	Passes arguments to the process. If a process is controlled by a script, the script is passed as an argument. Multiple arguments must be separated by a single space. A string cannot accommodate more than one space between arguments, nor allow for leading or trailing whitespace characters. Arguments must not exceed 80 characters (total). Type and dimension: string-scalar Example: "- a 8"
IgnoreArgs	A flag that indicates whether monitor ignores the argument list. ■ If the value is 0, it checks the process pathname and argument list. ■ If the value is 1, it only checks for the executable pathname and ignores the rest of the argument list. Type and dimension: boolean-scalar Default: 0

Resource type definition

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

Sample configurations

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

Phantom phantom_vxss (
    )

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

Debug log levels

The ProcessOnOnly agent uses the following debug log levels:

DBG_1

Zone agent

The Zone agent brings online, takes offline, monitors, and cleans Solaris 10 zones. You can use the agent to make zones highly available and to monitor them.

Note: Solaris 10 Update 3 or later enables attach and detach functionality for zones. Since the Zone agent supports this feature, you can patch a node where the service group that contains the zone resource is offline.

The ContainerOpts resource type attribute for this type has a default value of 0 for RunInContainer and a default value of 1 for PassCInfo. Symantec recommends that you do not change the values for these keys. Refer to the *Storage Foundation High Availability Virtualization Guide* for more information.

Dependencies

Typically no dependencies are required for the Zone resource, however if the zone root is on shared storage the resource may require the Mount and DiskGroup resources.

Figure 5-4 Sample service group that includes a Zone resource when the zone root is on shared storage with a loopback file system

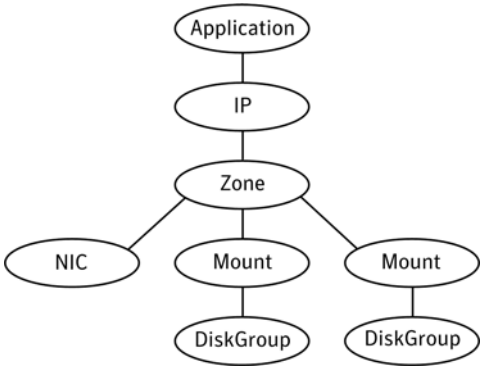
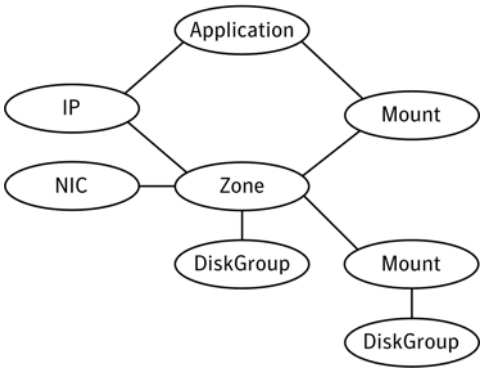


Figure 5-5 Sample service group that includes a Zone resource with the zone root on shared storage a direct mounted file system



Agent functions

Online	Brings a Solaris 10 non-global zone up and running.
Offline	Takes a Solaris 10 non-global zone down gracefully.
Monitor	Checks if the specified non-global zone is up and running.
Clean	A more forceful method for halting a Solaris 10 non-global zone.

Attributes

Table 5-12 Optional attributes

Optional attribute	Description
Pool	This is the resource pool name that is associated with the zone. Type and dimension: string-scalar

Table 5-12 Optional attributes

Optional attribute	Description
BootState	The value for the milestone service. Acceptable values follow: ■ single-user ■ multi-user ■ multi-user-server **Note:** Symantec recommends that you use the multi-user-server value for the BootState attribute. Type and dimension: string-scalar Default: multi-user
ShutDownGracePeriod	Specifies the interval in seconds from the Offline action to the execution of the shutdown within the zone. Type and dimension: integer-scalar Default: 0 Example: "10"
RunFsck	If the value of this attribute is 1, the Zone agent checks file system consistency for vxfs file systems. It uses the `fsck -y` command on all vxfs file systems that are defined in the zone's xml file. This file is located in /etc/zones. Adjust the Zone agent default OnlineTimeout value so that zone agent has sufficient time to run the `fsck` command before it brings the zone online. Type and dimension: boolean-scalar Default: 0 Example: 1
DetachZonePath	If disabled, the Zone agent skips detaching the Zone root during zone resource offline and clean. DetachZonePath is enabled (1) by default. Type and dimension: boolean-scalar Default: 1 Example: 0

Table 5-12 Optional attributes

Optional attribute	Description
ForceAttach	If disabled, the Zone agent attaches the ZonePath without the -F option during zone resource online. ForceAttach is enabled (1) by default. Type and dimension: boolean-scalar Default: 1 Example: 0

Resource type definition

```
type Zone (
    static str ArgList[] = { Pool, BootState,
    ShutdownGracePeriod, RunFsck, DetachZonePath, ForceAttach }
    static int ContainerOpts{} = { RunInContainer=0, PassCInfo=1 }
    str Pool
    str BootState = multi-user
    int ShutdownGracePeriod
    boolean RunFsck = 0
    boolean DetachZonePath = 1
    boolean ForceAttach = 1
)
```

Sample configurations

Application resource in a non-global zone for Solaris 10

In this example, configure a resource in a non-global zone: localzone1. The ZonePath of localzone1 is /zone1/root. The ContainerInfo attribute for this service group is set to ContainerInfo = { Name = “localzone1”, Type = “Zone”, Enabled = 1}. Configure the executable samba as StartProgram and StopProgram, with start and stop specified as command line arguments respectively. Configure the agent to monitor two processes: a process specified by the pid smbd.pid, and the process nmbd.

```
include "types.cf"

cluster vcszones_clus (
        UserNames = { admin = eHIaHChEIdIIgQIcHF,
                  z_zone_res_sys1 = dKLeIOjQJjJTjSKsHK }
        Administrators = { admin }
        )

system sys1 (
        )
```

```
system sys2 (
        )

group zoneapp_grp (
        SystemList = { sys1 = 0, sys2 = 1 }
        ContainerInfo = { Name = localzone1, Type = Zone, Enabled =
1 }
        AutoStartList = { sys1 }
        Administrators = { z_zone_res_sys1 }
        )

        Application samba_app (
        StartProgram = "/usr/sbin/samba start"
        StopProgram = "/usr/sbin/samba stop"
        PidFiles = { "/localzone1/root/var/lock/samba/smbd.pid" }
        MonitorProcesses = { "nmbd" }
        )

        Zone zone_res (
                )

samba_app requires zone_res

        // resource dependency tree
        //
        //      group grp_sol10zone
        //      {
        //     Application samba_app
        //                 {
        //                   Zone zone_res
        //                 }
        //      }
```

Debug log levels

The Zone agent uses the following debug log levels:

DBG_1, DBG_2, DBG_3, DBG_4, DBG_5

LDom agent

The LDom agent brings logical domains (LDoms) online, takes them offline, and monitors them. You can use this agent to monitor LDoms and to make them highly available.

For detailed information about support for logical domains, refer to the *Storage Foundation High Availability Virtualization Guide*.

Configuring primary and guest domain dependencies and failure policy

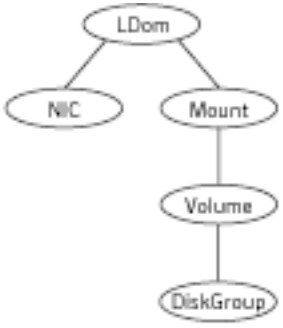
For all the guest domains that are configured in the cluster, the agent performs the following commands to set:

- The dependency between the primary and guest domains.
  ```
  # ldm set-domain master=primary guestldom
  ```
- The failure-policy of the primary domain to stop.
  ```
  # ldm set-domain failure-policy=stop primary
  ```

Dependencies

The LDom resource depends on the NIC resource. It also depends on a storage resource, such as Mount, Volume, Zpool, or Disk.

Figure 5-6 Sample service group for an LDom resource that monitors an image file



Network resources

Use the NIC agent to monitor the network adapter for the LDom, whether it is virtual or physical.

Storage resources

- Veritas Volume Manager (VxVM) exposed volumes
  Use the Volume and DiskGroup agents to monitor a VxVM volume.
- ZFS volumes
  Use the Zpool agent to monitor a ZFS volume.
- Image file
  - Image file in a volume that is managed by Veritas Volume Manager (VxVM)
    Use the Mount, Volume, and DiskGroup agents to monitor the image file.
  - Image file in an NFS share
    Use the Mount agent to monitor the image file.
  - Image file in a ZFS volume
    Use the Mount and Zpool agents to monitor the image file.
  - Image file in a partition of a physical disk
    Use the Mount and Disk agents to monitor the image file.

Agent functions

Online	Starts the LDom.
Offline	Stops the LDom.
Monitor	Monitors the status of the LDom.
Clean	Stops the LDom forcefully.

State definitions

ONLINE	Indicates that the LDom is up and running.
OFFLINE	Indicates that the LDom is down.
FAULTED	Indicates that the LDom is down when the VCS engine expects it to be up and running. If the MonitorCPU attribute is set to true (1), CPU usage of either 0% or 100% is interpreted as a fault.
UNKNOWN	Indicates the agent cannot determine the LDom's state. A configuration problem likely exists in the resource or the LDom.

Attributes

Table 5-13 Required attributes

Required attribute	Description
LDomName	The name of the LDom that you want to monitor. Type-dimension: string-scalar Default: n/a Example: "ldom1"

Table 5-14 Optional attributes

Optional attribute	Description
CfgFile	The absolute location of the XML file that contains the LDom configuration. The online agent function uses this file to create LDoms as necessary. Refer to the `ldm(1M)` man page for information on this file. To create the configuration file for an LDom, run the following command: `$ ldm list-constraints -x ldom_name > ldom_name.xml` The configuration file must be present locally on all of the systems or on a shared disk where it is accessible by all of the systems Type-dimension: string-scalar Default: n/a Example: "/root/ldom-cfg/ldom1.xml"

Table 5-14 Optional attributes

Optional attribute	Description
MonitorCPU	Specifies whether the LDom agent monitors the CPU usage of the LDom. If the CPU usage of all of the VCPUs attached to the LDom is equal to either 0% or 100%, then the resource is declared FAULTED. For an LDom with one VCPU, set this attribute to 0. This setting is to work around an LDom limitation where an LDom with one VCPU always reports CPU usage of 100%. Type-dimension: boolean-scalar Default: 1
NumCPU	The number of virtual CPUs that you want to attach to the LDom when it is online. If you set this attribute to a positive value, the agent detaches all of the VCPUs when the service group goes offline. Do not reset this value to 0 after setting it to 1. Type-dimension: integer-scalar Default: 0

Resource type definition

```
type LDom (
    static keylist RegList = { NumCPU }
    static str AgentFile = "bin/Script50Agent"
    static str ArgList[] = { LDomName, CfgFile, MonitorCPU, NumCPU }
    str LDomName
    str CfgFile
    boolean MonitorCPU = 1
    int NumCPU
)
```

Sample configuration

```
LDom ldom1 (
    LDomName = "ldom1"
)
```

Debug log levels

The LDom agent uses the following debug log levels:

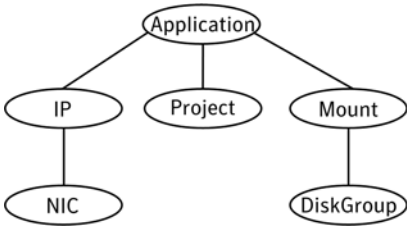
DBG_1, DBG_2, DBG_3, DBG_4, DBG_5

Project agent

The Project agent adds, deletes, and monitors Solaris projects. You can use the agent to make projects highly available or to monitor them.

Dependencies

Figure 5-7 Sample service group that includes a Project resource



Agent functions

Online	Creates a Solaris project, if one is not present. Modifies a Solaris project, if one present.
Offline	Deletes a Solaris project if the value of the OfflineDelProject attribute is 1. If the value of the OfflineDelProject attribute is 0, then the Solaris project is not deleted from the /etc/project file.
Monitor	Checks if the specified project is up and running.
Clean	Deletes a Solaris project if the value of the OfflineDelProject attribute is 1. If the value of the OfflineDelProject attribute is 0, then the Solaris project is not deleted from the /etc/project file.

Attributes

Table 5-15 Required attributes

Required attribute	Description
OfflineDelProject	Use the OfflineDelProject attribute to tell the Project agent to remove a project entry from the /etc/project file when the agent invokes the offline or clean agent functions. The OfflineDelProject's default value is 1, which instructs the Project agent to remove the project's entry from the /etc/project file. Set the value of the OfflineDelProject's attribute to 0, if you do not want the project to be deleted from the /etc/project file. Default: 1

Table 5-16 Optional attributes

Optional attribute	Description
User	Comma separated list of existing users that are part of the project. Type and dimension: string-scalar

Resource type definition

```
type Project (
    static str ArgList[] = { User, OfflineDelProject }
    static int ContainerOpts{} = { RunInContainer=0, PassCInfo=1 }
    static int FaultPropagation = 0
    str User
    boolean OfflineDelProject = 1
    )
```

Sample configuration

```
include "types.cf"
include "OracleTypes.cf"

cluster vcs (
    )
group grp_xrm (
    SystemList = { sysa = 0 }
```

```
    ContainerInfo @sysa = { Name = ora_project, Type = XRM,
    Enabled = 1 }
        )

        Oracle ora_res (
                Sid = oradb
                Owner = oracle
                Home = "/oraHome/app"
                )

        Project proj_res (
                User = oracle
                OfflineDelProject = 0
                )

ora_res requires proj_res
// resource dependency tree
//
//      group grp_xrm
//      {
//      Oracle ora_res
//          {
//          Project proj_res
//          }
//      }
```

Debug log levels

The Project agent uses the following debug log levels:

DBG_1

Chapter 6

Infrastructure and support agents

This chapter contains the following agents:

- "About the infrastructure and support agents" on page 229
- "NotifierMngr agent" on page 230
- "Proxy agent" on page 237
- "Phantom agent" on page 241
- "RemoteGroup agent" on page 243

About the infrastructure and support agents

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

NotifierMngr agent

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

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

Dependency

The NotifierMngr resource can depend on the NIC resource.

Agent functions

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

State definitions

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

Attributes

Table 6-1 Required attributes

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

Table 6-2 Optional attributes

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

Table 6-2 Optional attributes

Optional attribute	Description
MessagesQueue	Size of the VCS engine's message queue. Minimum value is 30. Type and dimension: integer-scalar Default: 30
NotifierListeningPort	Any valid, unused TCP/IP port number. Type and dimension: integer-scalar Default: 14144
NotifierSourceIP	If this attribute is populated, all the notifications sent from the notifier (SMTP and SNMP) will be sent from the interface having this IP address. **Note:** Make sure that the SourceIP given in this attribute is present in the /etc/hosts file or is DNS-resolvable. Type and dimension: string-scalar Example: "10.209.77.111"
SmtpFromPath	Set to a valid email address, if you want the notifier to use a custom email address in the FROM: field. Type and dimension: string-scalar Example: "usera@example.com"
SmtpRecipients	Specifies the email address where SMTP sends information and the severity level of the messages. The severity levels of messages are Information, Warning, Error, and SevereError. Specifying a given severity level for messages indicates that all messages of equal or higher severity are received. **Note:** SmtpRecipients is a required attribute if you specify SmtpServer. Type and dimension: string-association Example: "james@example.com" = SevereError, "admin@example.com" = Warning

Table 6-2 Optional attributes

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

Table 6-2 Optional attributes

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

Resource type definition

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

Sample configuration

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

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

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

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

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

Configuration

```
system north

system south

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

    Phantom my_phantom (
    )

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

group Grp1 (
    SystemList = { north = 0, south = 1 }
    AutoStartList = { north }
    )

    Proxy nicproxy(
```

```
TargetResName = "NicGrp_en0"
)

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

ntfr requires nicproxy

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

Debug log levels

The NotifierMngr agent uses the following debug log levels:

DBG_1, DBG_2, DBG_3, DBG_5

Proxy agent

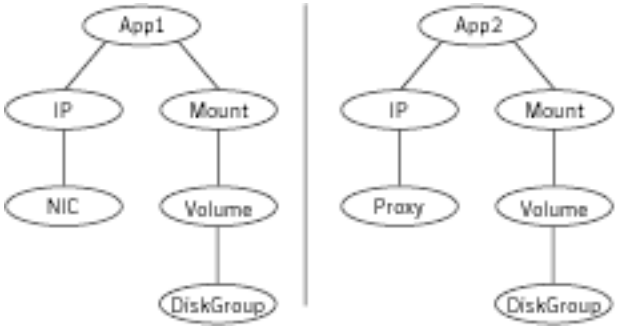
The Proxy agent mirrors the state of another resource on a local or remote system. It provides a means to specify and modify one resource and have its state reflected by its proxies. You can use the agent when you need to replicate the status of a resource.

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

Dependencies

No dependencies exist for the Proxy resource.

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



Agent functions

Monitor Determines status based on the target resource status.

Attributes

Table 6-3 Required attribute

Required attribute	Description
TargetResName	Name of the target resource that the Proxy resource mirrors. The target resource must be in a different resource group than the Proxy resource. Type and dimension: string-scalar Example: "tmp_VRTSvcs_file1"

Table 6-4 Optional attribute

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

Resource type definition

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

Sample configurations

Configuration 1

The proxy resource mirrors the state of the resource tmp_VRTSvcs_file1 on the local system.

```
Proxy proxy1 (
    TargetResName = "tmp_VRTSvcs_file1"
)
```

Configuration 2

The proxy resource mirrors the state of the resource tmp_VRTSvcs_file1 on sysa.

```
Proxy proxy1(
    TargetResName = "tmp_VRTSvcs_file1"
    TargetSysName = "sysa"
)
```

Configuration 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 in grp2; a target resource and its proxy cannot be in the same group.

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

    MultiNICA mnic (
        Device@sysa = { le0 = "166.98.16.103",qfe3 = "166.98.16.103"
}
        Device@sysb = { le0 = "166.98.16.104",qfe3 = "166.98.16.104"
}
        NetMask = "255.255.255.0"
        ArpDelay = 5
        Options = "failover"
        RouteOptions@sysa = "default 166.98.16.103 0"
```

```
        RouteOptions@sysb = "default 166.98.16.104 0"
        )
    IPMultiNIC ip1 (
        Address = "166.98.14.78"
        NetMask = "255.255.255.0"
        MultiNICResName = mnic
        Options = "failover"
        )

ip1 requires mnic

group grp2 (
    SystemList = { sysa = 0, sysb = 1 }
    AutoStartList = { sysa }
    )
    IPMultiNIC ip2 (
        Address = "166.98.14.79"
        NetMask = "255.255.255.0"
        MultiNICResName = mnic
        Options = "mtu m"
        )
Proxy proxy (
        TargetResName = mnic
        )
ip2 requires proxy
```

Debug log levels

The Proxy agent uses the following debug log levels:

DBG_1, DBG_2

Phantom agent

The Phantom agent enables VCS to determine the state of parallel service groups that do not include OnOff resources.

Do not use the Phantom resource in failover service groups.

Note: Do not attempt manual online or offline operations on the Phantom resource at the resource level. Do not use `hares` commands on the Phantom resource at the resource level. Unpredictable behavior results when you try a manual online or offline procedure or an `hares` command on a Phantom resource. You can perform commands on the service group that contains the Phantom resource.

Dependencies

No dependencies exist for the Phantom resource.

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

Agent functions

Monitor Determines status based on the status of the service group.

Resource type definition

```
type Phantom (
    static str ArgList[] = { Dummy }
    str Dummy

)
```

Sample configurations

Configuration 1

```
Phantom boo (
)
```

Configuration 2

The following example shows a complete main.cf, in which the FileNone resource and the Phantom resource are in the same group.

```
include "types.cf"

cluster PhantomCluster

system sysa (
    )

system sysb (
    )

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

    FileNone my_file_none (
        PathName = "/tmp/file_none"
        )
    Phantom my_phantom (
        )

    // resource dependency tree
    //
    //    group maingroup
    //    {
    //    Phantom my_Phantom
    //    FileNone my_file_none
    //    }
```

RemoteGroup agent

The RemoteGroup agent establishes dependencies between applications that are configured on different VCS clusters. For example, you configure an Apache resource in a local cluster, and a MySQL resource in a remote cluster. In this example, the Apache resource depends on the MySQL resource. You can use the RemoteGroup agent to establish this dependency between these two resources.

With the RemoteGroup agent, you can monitor or manage a service group that exists in a remote cluster. Some points about configuring the RemoteGroup resource follow:

- For each remote service group that you want to monitor or manage, you must configure a corresponding RemoteGroup resource in the local cluster.
- Multiple RemoteGroup resources in a local cluster can manage corresponding multiple remote service groups in different remote clusters.
- You can include the RemoteGroup resource in any kind of resource or service group dependency tree.
- A combination of the state of the local service group and the state of the remote service group determines the state of the RemoteGroup resource.

Symantec supports the RemoteGroup agent when:

- When it points to a global group
  The RemoteGroup agent must then map the state of the global group in the local cluster.
- When it is configured inside a local parallel service group
  The RemoteGroup resources on all cluster nodes monitor the same remote service group unless its attributes are localized.
- When it is configured inside a local failover service group

For more information on the functionality of this agent see the *Veritas Cluster Server Administrator's Guide*.

Dependency

As a best practice, establish a RemoteGroup resource dependency on a NIC resource. Symantec recommends that the RemoteGroup resource not be by itself in a service group.

Agent functions

Online	Brings the remote service group online. See the "ControlMode" on page 246 for more information.
Offline	Takes the remote service group offline. See the "ControlMode" on page 246 for more information.
Monitor	Monitors the state of the remote service group. The true state of the remote service group is monitored only on the online node in the local cluster. See the "VCSSysName" on page 245.
Clean	If the RemoteGroup resource faults, the Clean function takes the remote service group offline. See the "ControlMode" on page 246 for more information.

State definitions

ONLINE	Indicates that the remote service group is in an ONLINE state. If the ReturnIntOffline attribute is not set to RemotePartial, then the remote service group is either in an ONLINE or PARTIAL state. See "ReturnIntOffline" on page 249.
OFFLINE	Indicates that the remote service group is in an OFFLINE or FAULTED state. The true state of the remote service group is monitored only on the online node in the local cluster. The RemoteGroup resource returns intentional offline if the attribute ReturnIntOffline is set to an appropriate value. See "ReturnIntOffline" on page 249.
FAULTED	Indicates that the RemoteGroup resource has unexpectedly gone offline.
UNKNOWN	Indicates that a problem exists either with the configuration or the ability of the RemoteGroup resource to determine the state of the remote service group.

Attributes

Table 6-5 Required attributes

Required attribute	Description
IpAddress	The IP address or DNS name of a node in the remote cluster. The IP address can be either physical or virtual. When configuring a virtual IP address of a remote cluster, do not configure the IP resource as a part of the remote service group. Type and dimension: string-scalar Examples: "www.example.com" or "11.183.12.214"
Port	This is a required attribute when the remote cluster listens on a port other than the default value of 14141. See "Port" on page 248.
GroupName	The name of the service group on the remote cluster that you want the RemoteGroup agent to monitor or manage. Type and dimension: string-scalar Example: "DBGrp"
VCSSysName	You must set this attribute to either the VCS system name or the ANY value. ■ ANY The RemoteGroup resource goes online if the remote service group is online on any node in the remote cluster. ■ *VCSSysName* Use the name of a VCS system in a remote cluster where you want the remote service group to be online when the RemoteGroup resource goes online. Use this to establish a one-to-one mapping between the nodes of the local and remote clusters. Type and dimension: string-scalar Example: "vcssys1" or "ANY"

Table 6-5 Required attributes

Required attribute	Description
ControlMode	Select only one of these values to determine the mode of operation of the RemoteGroup resource: MonitorOnly, OnlineOnly, or OnOff. ■ OnOff The RemoteGroup resource brings the remote service group online or takes it offline. When you set the VCSSysName attribute to ANY, the SysList attribute of the remote service group determines the node where the remote service group onlines. ■ MonitorOnly The RemoteGroup resource only monitors the state of the remote service group. The RemoteGroup resource cannot online or offline the remote service group. Make sure that you bring the remote service group online before you online the RemoteGroup resource. ■ OnlineOnly The RemoteGroup resource only brings the remote service group online. The RemoteGroup resource cannot take the remote service group offline. When you set the VCSSysName attribute to ANY, the SysList attribute of the remote service group determines the node where the remote service group onlines. Type and dimension: string-scalar

Table 6-5 Required attributes

Required attribute	Description
Username	This is the login user name for the remote cluster. When you set the ControlMode attribute to OnOff or OnlineOnly, the Username must have administrative privileges for the remote service group that you specify in the GroupName attribute. When you use the RemoteGroup Wizard to enter your username data, you need to enter your username and the domain name in separate fields. For a cluster that has the Symantec Product Authentication Service, you do not need to enter the domain name. For a secure remote cluster: ■ Local Unix user user@nodename—where the nodename is the name of the node that is specified in the IpAddress attribute. Do not set the DomainType attribute. ■ NIS or NIS+ user user@domainName—where domainName is the name of the NIS or NIS+ domain for the user. You must set the value of the DomainType attribute to either to nis or nisplus. Type and dimension: string-scalar Example: ■ For a cluster without the Symantec Product Authentication Service: "johnsmith" ■ For a secure remote cluster: "foobar@example.com"
Password	This is the password that corresponds to the user that you specify in the Username attribute. You must encrypt the password with the `vcsencrypt -agent` command. **Note:** Do not use the vcsencrypt utility when entering passwords from a configuration wizard or the Cluster Manager (Java Console). Type and dimension: string-scalar

Table 6-6 Optional attributes

Optional attribute	Description
DomainType	For a secure remote cluster only, enter the domain type information for the specified user. For users who have the domain type unixpwd, you do not have to set this attribute. Type: string-scalar Example: "nis", "nisplus"
BrokerIp	For a secure remote cluster only. If you need the RemoteGroup agent to communicate to a specific authentication broker, set the value of this attribute to the broker's IP address. Type: string-scalar Example: "128.11.295.51"
Port	The port where the remote engine listens for requests. This is an optional attribute, unless the remote cluster listens on a port other than the default value of 14141. Type and dimension: integer-scalar Default: 14141
OfflineWaitTime	The maximum expected time in seconds that the remote service group may take to offline. VCS calls the clean function for the RemoteGroup resource if the remote service group takes a longer time to offline than the time that you have specified for this attribute. Type and dimension: integer-scalar Default: 0

Table 6-6 Optional attributes

Optional attribute	Description
ReturnIntOffline	Select one of the following values for RemoteGroup to return IntentionalOffline: ■ RemotePartial—Indicates that the RemoteGroup resource returns an IntentionalOffline if the remote service group is in an ONLINE|PARTIAL state. ■ RemoteOffline—Indicates that the RemoteGroup resource returns an IntentionalOffline if the remote service group is in an OFFLINE state. ■ RemoteFaulted—Indicates that the RemoteGroup resource returns an IntentionalOffline if the remote service group is OFFLINE|FAULTED. You can use these values in combinations with each other. You must set the IntentionalOffline attribute of the RemoteGroup resource type to 1 for this attribute to work properly. For more information about this attribute, see the *Veritas Cluster Server Administrator's Guide*. Type and dimension: string-vector Default: ""
OfflineMonitoringNode	Defines the cluster node that performs the offline monitoring of the remote service group. This is an internal attribute. Do not modify.

Table 6-7 Type-level attributes

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

Table 6-7 Type-level attributes

Type level attributes	Description
ToleranceLimit MonitorInterval	If you expect the RemoteGroup agent to tolerate sudden offlines of the remote service group, then modify the ToleranceLimit attribute. See the *Veritas Cluster Server Administrator's Guide* for more information about these attributes.
ExternalStateChange	If you want the local service group to go online or offline when the RemoteGroup resource goes online or offline outside VCS control, set the attribute ExternalStateChange appropriately. See the *Veritas Cluster Server Administrator's Guide* for more information about these attributes.

Resource type definition

```
type RemoteGroup (
    static int OnlineRetryLimit = 2
    static int ToleranceLimit = 1
    static boolean IntentionalOffline = 1
    static str ArgList[] = { IpAddress, Port, Username, Password,
    GroupName, VCSSysName, ControlMode, OfflineWaitTime,
    DomainType, BrokerIp, ReturnIntOffline }
    str IpAddress
    int Port = 14141
    str Username
    str Password
    str GroupName
    str VCSSysName
    str ControlMode
    int OfflineWaitTime
    str DomainType
    str BrokerIp
    str ReturnIntOffline[] = {}
    temp str OfflineMonitoringNode
)
```

Debug log levels

The RemoteGroup agent uses the following debug log levels:

DBG_1

Chapter 7

Testing agents

This chapter contains the following agents:

- "About the testing agents" on page 251
- "ElifNone agent" on page 252
- "FileNone agent" on page 254
- "FileOnOff agent" on page 256
- "FileOnOnly agent" on page 258

About the testing agents

Use the testing agents to provide high availability for program support resources. These resources are useful for testing service groups.

ElifNone agent

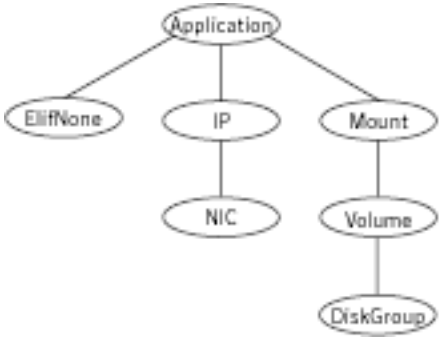
The ElifNone agent monitors a file. It checks for the file's absence.

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

Dependencies

No dependencies exist for the ElifNone resource.

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



Agent function

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

State definitions

ONLINE	Indicates that the file specified in the PathName attribute does not exist.
FAULTED	Indicates that the file specified in the PathName attribute exists.
UNKNOWN	Indicates that the value of the PathName attribute does not contain a file name.

Attributes

Table 7-1 Required attribute

Required attribute	Description
PathName	Specifies the complete pathname. Starts with a slash (/) preceding the file name. Type and dimension: string-scalar Example: "/tmp/file01"

Resource type definition

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

Sample configuration

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

Debug log levels

The ElifNone agent uses the following debug log levels:

DBG_1, DBG_4, DBG_5

FileNone agent

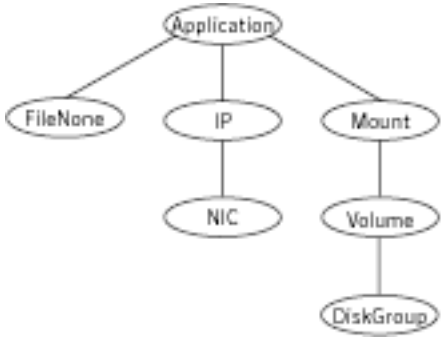
Monitors a file—checks for the file's existence.

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

Dependencies

No dependencies exist for the FileNone resource.

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



Agent functions

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

State definitions

ONLINE	Indicates that the file specified in the PathName attribute exists.
FAULTED	Indicates that the file specified in the PathName attribute does not exist.
UNKNOWN	Indicates that the value of the PathName attribute does not contain a file name.

Attribute

Table 7-2 Required attribute

Required attribute	Description
PathName	Specifies the complete pathname. Starts with a slash (/) preceding the file name. Type and dimension: string-scalar Example: "/tmp/file01"

Resource type definition

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

Sample configuration

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

Debug log levels

The FileNone agent uses the following debug log levels:

DBG_1, DBG_4, DBG_5

FileOnOff agent

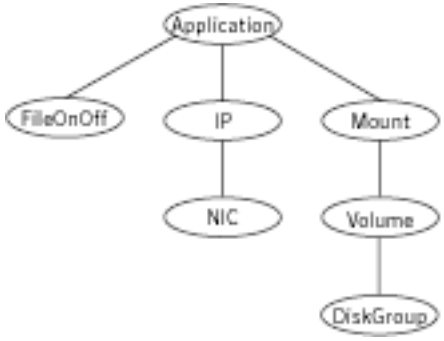
The FileOnOff agent creates, removes, and monitors files.

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

Dependencies

No dependencies exist for the FileOnOff resource.

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



Agent functions

Online	Creates an empty file with the specified name if the file does not already exist.
Offline	Removes the specified file.
Monitor	Checks for the specified file. If it exists, the agent reports as ONLINE. If it does not exist, the agent reports as OFFLINE.
Clean	Terminates all ongoing resource actions and takes the resource offline, forcibly when necessary.

State definitions

ONLINE	Indicates that the file specified in the PathName attribute exists.
OFFLINE	Indicates that the file specified in the PathName attribute does not exist.

FAULTED — Indicates that the file specified in the PathName attribute has been removed out of VCS control.

UNKNOWN — Indicates that the value of the PathName attribute does not contain a file name.

Attribute

Table 7-3 Required attribute

Required attribute	Description
PathName	Specifies the complete pathname. Starts with a slash (/) preceding the file name. Type and dimension: string-scalar Example: "/tmp/file01"

Resource type definition

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

Sample configuration

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

Debug log levels

The FileOnOff agent uses the following debug log levels:

DBG_1, DBG_4, DBG_5

FileOnOnly agent

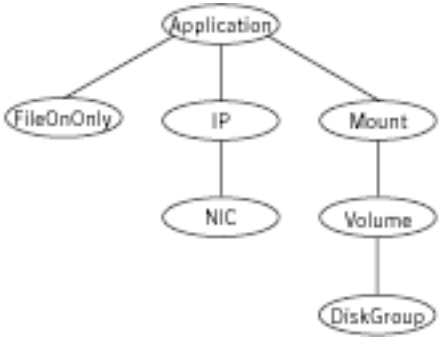
The FileOnOnly agent creates and monitors files.

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

Dependencies

No dependencies exist for the FileOnOnly resource.

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



Agent functions

Online	Creates an empty file with the specified name, unless one already exists.
Monitor	Checks for the specified file. If it exists, the agent reports as ONLINE. If it does not exist, the resource faults.

State definitions

ONLINE	Indicates that the file specified in the PathName attribute exists.
OFFLINE	Indicates that the file specified in the PathName attribute does not exist and VCS has not attempted to bring the resource online.
FAULTED	Indicates that the file specified in the PathName attribute has been removed out of VCS control.

UNKNOWN Indicates that the value of the PathName attribute does not contain a file name.

Attribute

Table 7-4 Required attributes

Required attribute	Description
PathName	Specifies the complete pathname. Starts with a slash (/) preceding the file name. Type and dimension: string-scalar Example: "/tmp/file02"

Resource type definition

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

Sample configuration

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

Debug log levels

The FileOnOnly agent uses the following debug log levels:

DBG_1, DBG_4, DBG_5

Glossary

administrative IP address
The operating system controls these IP addresses and brings them up even before VCS brings applications online. Use them to access a specific system over the network for doing administrative tasks, for example: examining logs to troubleshoot issues, cleaning up temp files to free space, etc. Typically, you have one administrative IP address per node.

agent function
Agent functions start, stop, fault, forcibly stop, and monitor resources using scripts. Sometimes called an entry point.

base IP address
The administrative IP address of the system.

entry point
See agent function.

floating IP address
See virtual IP address.

logical IP address
Any IP address assigned to a NIC.

NIC bonding
Combining two or more NICs to form a single logical NIC, which creates a fatter pipe.

operation
All agents have scripts that turn the resource on and off. Operations determine the action that the agent passes to the resource. See None operation, OnOff operation, and OnOnly operation.

None operation
For example the NIC resource. Also called persistent resource, this resource is always on. This kind of resource has no online and offline scripts, and only monitors a resource.

OnOff operation
For example the IP and Share agents--in fact most agents are OnOff. This resource has online and offline scripts. Often this type of resource does not appear in the types file because by default when a resource does not have this resource type defined, it is OnOff.

OnOnly operation
For example the NFS, FileOnOnly resources. This kind of resource has an online script, but not an offline one.

plumb
Term for enabling an IP address—used across all platforms in this guide.

test IP address

IP addresses to help determine the state of a link by sending out a ping probe to another NIC (on another system.) Requires a return ping to complete the test. Test IP addresses can be the same as base IP addresses.

virtual IP address

IP addresses that can move from one NIC to another or from one node to another. VCS fails over these IP address with your application. Sometimes called a floating IP address.

Index

Numerics

802.1Q trunking 81

A

about
- Network agents 79
- Samba agents 166

agent
- modifying 20

agent functions
- Apache Web server agent 183
- Application agent 196
- CoordPoint agent 204
- Disk agent 47
- DiskGroup agent 24
- DiskGroupSnap agent 35
- DNS agent 135
- ElifNone agent 252
- FileNone agent 254
- FileOnOff agent 256
- FileOnOnly agent 258
- IP agent 83
- IPMultiNIC agent 97
- IPMultiNICB agent 113
- LDom agent 222
- Mount agent 56
- MultiNICA agent 103
- MultiNICB agent 120
- NetBIOS agent 176
- NFS agent 149
- NFSRestart agent 154
- NIC agent 90
- NotifierMngr agent 230
- Phantom agent 241
- Process agent 209
- ProcessOnOnly agent 213
- Project agent 226
- Proxy agent 237
- RemoteGroup agent 244
- SambaServer agent 168
- SambaShare agent 173
- Share agent 162
- Volume agent 49
- VolumeSet agent 52
- Zone agent 217
- Zpool agent 73

agents
- Apache Web server 182
- Application 194
- CoordPoint 204
- Disk 47
- DiskGroup 24
- DiskGroupSnap 34
- DNS 134
- ElifNone 252
- FileNone 254
- FileOnOff 256
- FileOnOnly 258
- IP 82
- IPMultiNIC 96
- IPMultiNICB 112
- LDom 221
- Mount 55
- MultiNICA 102
- MultiNICB 119
- NetBIOS 176
- NFS 148
- NFSRestart 153
- NIC 89
- NotifierMngr 230
- Phantom 241
- Process 208
- ProcessOnOnly 213
- Project 226
- Proxy 237
- RemoteGroup 243
- SambaServer 168
- SambaShare 173
- Share 162
- Volume 49
- VolumeSet 52
- Zone 216
- Zpool 72

agents, typical functions 19
Apache Web server agent
    agent functions 183
    attributes 184
    description 182
    detecting application failure 190
    sample configuration 191
    state definitions 183
Application agent
    agent functions 196
    attributes 198
    description 194
    high availability fire drill 194
    resource type definition 201
    sample configurations 202
    state definitions 197
association dimension 20
attribute data types 20
attributes
    Application agent 198
    Base and Mpathd modes 122
    Base mode 123
    CoordPoint agent 205
    Disk agent 48
    DiskGroup agent 27
    DiskGroupSnap agent 36
    DNS agent 137
    ElifNone agent 253
    FileNone agent 255
    FileOnOff agent 257
    FileOnOnly agent 259
    IP agent 85
    IPMultiNIC agent 98
    IPMultiNICB agent 115
    LDom agent 223
    Mount agent 59
    MultiNICA agent 104
    MultiNICB agent 121
    Multipathing mode 126
    NFS agent 149
    NFSRestart agent 156
    NIC agent 91
    NotifierMngr agent 231
    ProcessOnOnly agent 214
    Proxy agent 238
    RemoteGroup agent 245
    SambaServer agent 170
    Share agent 164
    Volume agent 50
    VolumeSet agent 53
    Zpool agent 75
attributes, modifying 19, 20

B

Base mode 128
boolean data types 20
bundled agents 19

C

Checklist to ensure the proper operation of MultiNICB 111
Cluster Manager (Java Console), modifying attributes 20
CNAME record 142
configuration files
    main.cf 242
    modifying 20
    types.cf 19
configuring, Samba agents 167
CoordPoint agent
    agent functions 204
    attributes 205
    description 204
    resource type definition 206
    sample configurations 207
    state definitions 205

D

data type
    boolean 20
    string 20
data types
    integer 20
description, resources 19
dimensions
    keylist 20
    scalar 20
    vector 20
Disk agent
    agent functions 47
    attributes 48
    description 47
    resource type definition 48
    state definitions 47
DiskGroup agent
    agent functions 24

attributes 27
description 24
high availability fire drill 31
resource type definition 31
sample configurations 33
state definitions 26
DiskGroupSnap agent
agent functions 35
attributes 36
description 34
resource type definition 39
sample configurations 40
state definitions 35
DNS agent 136
agent functions 135
attributes 137
description 134
resource type definition 141
sample web server configuration 142

E

ElifNone agent
agent functions 252
attributes 253
description 252
resource type definition 253
sample configuration 253
state definitions 252

F

Fiber Channel adapter 33
FileNone agent
agent functions 254
attribute 255
description 254
resource type definition 255
sample configurations 255
state definitions 254
FileOnOff agent
agent functions 256
attribute 257
description 256
state definitions 256
FileOnOnly agent
agent functions 258
attribute 259
description 258
resource type definition 259
sample configuration 259
state definitions 258

H

high availability fire drill 31, 67, 82, 89, 142, 157, 194, 208

I

integer data types 20
Interface configuration 131
IP agent
agent functions 83
attributes 85
description 82
high availability fire drill 82
resource type definitions 87
sample configurations 88
state definitions 84
IPMultiNIC agent
agent functions 97
attributes 98
description 96
resource type definitions 99
sample configuration 100
state definitions 97
IPMultiNICB agent 118
agent functions 113
description 112
manually migrating IP address 118
requirements 113
resource type definition 117
Solaris attributes 115
state definitions 114

K

keylist dimension 20

L

LDom agent
agent functions 222
attributes 223
description 221
resource type definition 224
sample configurations 224
state definitions 222

M

main.cf 19, 242
main.xml 19
modifying
    configuration files 20
modifying agents 20
monitor scenarios, DNS agent 142
Mount agent
    agent functions 56, 58
    attributes 59
    description 55
    high availability fire drill 67, 142, 157
    notes 67
    resource type definition 66
    sample configurations 71
MultiNICA agent
    agent functions 103
    attributes 104
    description 102
    resource type attributes 107
    RouteOptions, HP-UX 108
    RouteOptions, Solaris 108
    sample configurations 109
    state definitions 103
MultiNICB agent
    administrative IP addresses 131
    agent functions 120
    attributes 121
    base and multipath, Solaris 128
    description 119
    resource type definition 128
    sample configurations 131
    state definitions 121
    test IP addresses 131
Multipathing mode 130

N

NetBIOS agent
    agent functions 176
    description 176
    resource type definition 177
    sample configurations 179
    state definitions 177
NFS agent
    agent functions 149
    attributes 149
    description 148
    resource type definition 151
    sample configurations 152
    state definitions 149
NFSRestart agent
    agent functions 154
    attributes 156
    description 153
    resource type definition 157
    sample configuration 160
    state definitions 155
NIC agent
    agent functions 90
    attributes 91
    description 89
    high availability fire drill 89
    resource type definitions 92
    sample configurations 93
    state definitions 90
noautoimport flag 32
NotifierMngr agent
    agent functions 230
    attributes 231
    description 230
    resource type definition 234
    sample configurations 235
    state definitions 230

O

online query 142

P

Phantom agent
    agent functions 241
    description 241
    resource type definition 241
    sample configurations 241
prerequisites
    Samba agents 166
Process agent 211
    agent functions 209
    attributes 211
    description 208
    high availability fire drill 208
    resource type definition 211
    sample configurations 212
    state definitions 210
ProcessOnOnly agent
    agent functions 213
    attributes 214

description 213
resource type definition 215
sample configurations 215
state definitions 213
Project agent
agent functions 226
attributes 227
description 226
resource type definition 227
sample configuration 227
Proxy agent
agent functions 237
attributes 238
description 237
resource type definition 239
sample configurations 239

R

RemoteGroup agent
agent functions 244
attributes 245
description 243
resource type definition 250
state definitions 244
resource type definition 51
SambaShare agent 174
resource type definitions
Application agent 201
CoordPoint agent 206
Disk agent 48
DiskGroup agent 31
DiskGroupSnap agent 39
DNS agent 141
ElifNone agent 253
FileNone agent 255
FileOnOnly agent 259
IP agent 87
IPMultiNIC agent 99
IPMultiNICB agent 117
LDom agent 224
Mount agent 66
MultiNICA agent 107
MultiNICB agent 128
NetBIOS agent 177
NFS agent 151
NFSRestart agent 157
NIC agent 92
NotifierMngr agent 234
Phantom agent 241
Process agent 211
ProcessOnOnly agent 215
Project agent 227
Proxy agent 239
RemoteGroup agent 250
SambaServer agent 172
Share agent 164
Volume agent 51
VolumeSet agent 53
Zone agent 219
Zpool agent 76
resource types 19
resources
description of 19

S

Samba agents 166
overview 166
prerequisites 166
Samba agents configuring 167
SambaServer agent
agent functions 168
attributes 170
description 168
resource type definition 172
sample configuration 172
state definitions 169
SambaShare agent 173
agent functions 173
attributes 174
resource type definition 174
sample configurations 175
state definitions 173
sample configurations 118
Apache Web server agent 191
Application agent 202
CoordPoint agent 207
DiskGroup agent 33
DiskGroupSnap agent 40
ElifNone agent 253
FileNone agent 255
FileOnOff agent 257
FileOnOnly agent 259
IP agent 88
IPMultiNIC 100
IPMultiNICB agent 118
LDom agent 224
Mount agent 71
MultiNICA agent 109

MultiNICB agent 131
NetBIOS agent 179
NFS agent 152
NFSRestart agent 160
NIC agent 93
NotifierMngr agent 235
Phantom agent 241
Process agent 212
ProcessOnOnly agent 215
Project agent 227
Proxy agent 239
SambaServer agent 172
SambaShare agent 175
Share agent 165
Volume agent 51
Zpool agent 76
scalar dimension 20
secure DNS update 143
Share agent
agent functions 162
attributes 164
description 162
resource type definitions 164
sample configurations 165
state definitions 163
Solaris 1
state definitions 52, 136
Apache Web server agent 183
Application agent 197
CoordPoint agent 205
Disk agent 47
DiskGroup agent 26
DiskGroupSnap agent 35
DNS agent 136
ElifNone agent 252
FileNone agent 254
FileOnOff agent 256
FileOnOnly agent 258
IP agent 84
IPMultiNIC agent 97
IPMultiNICB agent 114
LDom agent 222
Mount agent 58
MultiNICA agent 103
MultiNICB agent 121
NetBIOS agent 177
NFS agent 149
NFSRestart agent 155
NIC agent 90
NotifierMngr agent 230
Process agent 210
ProcessOnOnly agent 213
RemoteGroup agent 244
SambaServer agent 169
SambaShare agent 173
Share agent 163
Volume agent 49
VolumeSet agent 52
Zpool agent 74
string data type 20

T

trigger script 130
trunking 81
types.cf 19

V

VCS, resource types 19
vector dimension 20
Volume agent
agent functions 49
attributes 50
description 49
sample configurations 51
state definitions 49
volume sets, Solaris 32
VolumeSet agent 52
agent functions 52
attributes 53
description 52
resource type definition 53

Z

Zone agent
agent functions 217
attributes 217
description 216
resource type definition 219
zones
resource attributes 21
Zpool agent
agent functions 73
attributes 75
description 72
resource type definition 76
sample configurations 76

state definitions 74