Cluster Server 7.4.1 Bundled Agents Reference Guide - Windows

VERITAS™

Last updated: 2019-09-17

Legal Notice

Copyright © 2019 Veritas Technologies LLC. All rights reserved.

Veritas and the Veritas Logo are trademarks or registered trademarks of Veritas Technologies LLC or its affiliates in the U.S. and other countries. Other names may be trademarks of their respective owners.

This product may contain third-party software for which Veritas is required to provide attribution to the third-party (“Third-Party Programs”). Some of the Third-Party Programs are available under open source or free software licenses. The License Agreement accompanying the Software does not alter any rights or obligations you may have under those open source or free software licenses. Refer to the third-party legal notices document accompanying this Veritas product or available at:

https://www.veritas.com/about/legal/license-agreements

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 Veritas Technologies LLC 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. VERITAS TECHNOLOGIES LLC 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, et seq. "Commercial Computer Software and Commercial Computer Software Documentation," as applicable, and any successor regulations, whether delivered by Veritas as on premises or hosted services. 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.

Veritas Technologies LLC
500 E Middlefield Road
Mountain View, CA 94043

http://www.veritas.com

Technical Support

Technical Support maintains support centers globally. All support services will be delivered in accordance with your support agreement and the then-current enterprise technical support policies. For information about our support offerings and how to contact Technical Support, visit our website:

https://www.veritas.com/support

You can manage your Veritas account information at the following URL:

https://my.veritas.com

If you have questions regarding an existing support agreement, please email the support agreement administration team for your region as follows:

Worldwide (except Japan)	CustomerCare@veritas.com
Japan	CustomerCare_Japan@veritas.com

Documentation

Make sure that you have the current version of the documentation. Each document displays the date of the last update on page 2. The latest documentation is available on the Veritas website:

https://sort.veritas.com/documents

Documentation feedback

Your feedback is important to us. Suggest improvements or report errors or omissions to the documentation. Include the document title, document version, chapter title, and section title of the text on which you are reporting. Send feedback to:

infoscaledocs@veritas.com

You can also see documentation information or ask a question on the Veritas community site:

http://www.veritas.com/community/

Veritas Services and Operations Readiness Tools (SORT)

Veritas Services and Operations Readiness Tools (SORT) is a website that provides information and tools to automate and simplify certain time-consuming administrative tasks. Depending on the product, SORT helps you prepare for installations and upgrades, identify risks in your datacenters, and improve operational efficiency. To see what services and tools SORT provides for your product, see the data sheet:

https://sort.veritas.com/data/support/SORT_Data_Sheet.pdf

Contents

Chapter 1 Introducing VCS bundled agents 11

About VCS bundled agents 11
Resources and their attributes 12
Modifying agents and their resources 12
Attributes 12
Attribute data types 13
Attribute dimensions 13
Categories of bundled agents 14
Storage agents 14
Network agents 14
FileShare agents 15
Services and applications agents 15
VCS infrastructure and support agents 15
Which agents should I use? 16

Chapter 2 Storage agents 18

About Storage agents for VCS 18
DiskReservation (DiskRes) agent 19
Agent functions 19
State definitions 20
Resource type definition 20
Attribute definitions 20
Sample configuration 21
Mount agent 22
Dependency 22
Agent functions 23
State definitions 23
Resource type definition 23
Attribute definitions 23
Sample configurations 25
Volume Manager Diskgroup (VMDg) agent 26
Agent functions 26
State definitions 27
Resource type definition 27
Attribute definitions 27

Sample configuration 30
MountV agent 30
Dependency 31
Agent functions 31
State definitions 31
Resource type definition 32
Attribute definitions 32
Sample configuration 36
VVRSnap agent 36
Agent functions 36
State definitions 37
Resource type definition 37
Attribute definitions 37
Sample configuration 38
VMwareDisks agent 38
Agent functions 39
State definitions 39
Resource type definition 39
Attributes 40
Sample configuration 42
VMNSDg agent 43
Agent functions 43
State definitions 43
Resource type definition 43
Attribute definitions 44
Sample configuration 46
NativeDisks agent 46
Agent functions 47
State definitions 47
Resource type definition 47
Attribute definitions 47
Sample configuration 47
AWS EBSVol agent 48
AzureDisk agent 52
GoogleDisk agent 56

Chapter 3 Network agents 61

About network agents 61
NIC agent 61
Agent functions 62
State definitions 62
Resource type definition 63

Attribute definitions 63
Sample configuration 64
IP agent 65
Dependency 65
Agent functions 65
State definitions 65
Resource type definitions 66
Attribute definitions 66
Sample configuration 67
IPv6 agent 67
Dependency 68
Agent functions 68
State definitions 68
Resource type definition 69
Attribute definitions 69
Sample configuration 70
Lanman agent 70
Dependency 71
Agent functions 71
State definitions 71
Resource type definition 72
Attribute definitions 72
Sample configurations 80
AWSIP agent 82
AzureIP agent 87
AzureDNSZone agent 93
Prerequisites 93
Dependencies 94
Agent functions 94
State definitions 95
Attributes 95
Resource type definition 96
Samples configurations 97
Delegating a domain to Azure DNS 98
GoogleIP agent 99

Chapter 4 FileShare agents 106

About FileShare agents 106
FileShare agent 106
Dependencies 107
Agent functions 107
State definitions 108

About ForceControl function 108
Resource type definition 108
Attribute definitions 109
Sample configuration 115
CompositeFileShare agent 116
Dependencies 117
Agent functions 117
State definitions 117
About ForceControl function 118
Resource type definition 118
Attribute definitions 119
Sample configuration 124

Chapter 5 Services and applications agents 126

About services and application agents 126
Internet Information Services agent 127
Agent functions 127
State definitions 128
Resource type definition 128
Attribute definitions 128
Sample configurations 130
GenericService agent 134
Agent functions 135
State definitions 135
Resource type definition 135
Attribute definitions 136
Sample configuration 138
ServiceMonitor agent 138
Agent functions 139
State definitions 139
Resource type definition 140
Attribute definitions 140
Sample configurations 142
Process agent 143
Agent functions 143
State definitions 143
Resource type definition 144
Attribute definitions 144
Sample configuration 148
MSMQ agent 149
Dependencies 149
Agent functions 149

State definitions 150
Resource type definition 150
Attribute definitions 150
Sample configuration 151
AzureAuth agent 151

Chapter 6 VCS infrastructure and support agents 156

About VCS infrastructure and support agents 156
NotifierMngr agent 157
Agent functions 157
State definitions 158
Resource type definition 158
Attribute definitions 159
Sample configuration 160
Registry Replication (RegRep) agent 161
Dependency 162
Agent functions 162
State definitions 162
Resource type definition 163
Attribute definitions 163
Sample configuration 167
Proxy agent 169
Agent functions 169
State definitions 169
Resource type definition 169
Attribute definitions 170
Sample configurations 170
Phantom agent 171
Agent functions 171
State definitions 171
Attribute definitions 171
Sample configuration 171
FileNone agent 172
Agent functions 172
State definitions 172
Resource type definition 172
Attribute definitions 173
Sample configuration 173
ElifNone agent 173
Agent functions 173
State definitions 173
Resource type definition 173

Attribute definitions 174
Sample configuration 174
FileOnOff agent 174
Agent functions 174
State definitions 174
Resource type definition 175
Attribute definitions 175
Sample configuration 175
FileOnOnly agent 175
Agent functions 175
State definitions 175
Resource type definition 176
Attribute definitions 176
Sample configuration 176
RemoteGroup agent 176
Dependency 177
Agent functions 177
State definitions 178
Resource type definition 178
Attribute definitions 179
Sample scenario 184

Chapter 7 VCS hardware replication agent for NetApp 187

About hardware replication agents for NetApp 187
NetApp Filer agent 188
Agent function 188
Resource type definition 188
Attribute definitions 188
Sample configuration 189
NetApp SnapDrive agent 189
Dependency 189
Agent functions 189
Resource type definition 190
Attribute definitions 190
Sample configuration 191
NetApp SnapMirror agent 191
Dependency 191
Agent functions 192
Action function 192
Resource type definition 193
Attribute definitions 194
Sample configuration 195

Chapter 8 Troubleshooting bundled agents 196

About troubleshooting bundled agents 196
VCS logging 197
Troubleshooting storage agents 197
Disk Reservation agent error messages 197
Mount agent error messages 202
Volume Manager (VMDg) agent error messages 208
MountV agent error messages 210
Troubleshooting network agents 213
NIC agent error messages 213
IP agent error messages 215
Lanman agent error messages 216
Troubleshooting FileShare agents 219
FileShare agent error messages 220
CompositeFileShare agent error messages 221
Troubleshooting services and application agents 222
IIS agent error messages 222
GenericService agent error messages 225
ServiceMonitor agent error messages 228
Process agent error messages 231
MSMQ agent error messages 233
Troubleshooting infrastructure and support agents 234
Notifier Manager agent error messages 234
Registry Replication agent error messages 236
Proxy agent error messages 237
Phantom agent error message 238
FileNone agent error message 238
ElifNone agent error message 238
FileOnOff agent error messages 238
FileOnOnly agent error messages 238
Troubleshooting NetApp agents 239

Chapter 1

Introducing VCS bundled agents

This chapter includes the following topics:

- About VCS bundled agents
- Resources and their attributes
- Modifying agents and their resources
- Attributes
- Categories of bundled agents
- Which agents should I use?

About VCS bundled agents

Bundled agents are VCS processes that manage resources of predefined resource types. The agents manage resources according to commands received from the VCS engine, HAD. Bundled agents are a part of VCS, and are installed when InfoScale Availability or InfoScale Enterprise is installed. A system 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 is started, it obtains the necessary configuration information from VCS. It then periodically monitors the resources, and updates VCS with the resource status.

Agents typically:

- Bring resources online.
- Take resources offline.

- Monitor resources and report state changes to VCS.

This document describes the VCS bundled agents along with their resource type definitions, attribute definitions, and sample configurations. The chapter on configuring resources and applications in the *Cluster Server Administrator's Guide* describes how to configure these agents.

Resources and their attributes

Resources are the key parts of a system and are known by their type, such as a disk, a volume, or an IP address. Resource types are defined in the types.cf file by a collection of attributes. The VCS configuration file, main.cf, contains the values for the resource attributes and has an include directive to the types.cf file.

There are three categories of resources: On-Off, On-Only, and Persistent. The category of a resource governs the functions that the corresponding agent performs. For example, if the resources of an agent are of type On-Off, the agent could online and offline the resources. If the resources of an agent are of type On-Only, the agent could online the resources but cannot offline them. For persistent resources, the agent can neither online nor offline a resource.

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 monitors an IP address resource. The specific address to be monitored is identified by the attribute Address whose value is the specific IP address.

Modifying agents and their resources

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

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

Attributes

Configure VCS components with attributes. Attributes contain data about the cluster, systems, service groups, resources, resource types, agent, and heartbeats—if you want to use global clusters. Each attribute has a definition and a value. Some attributes also have default values.

Attribute data types

The following table summarizes the attribute data types for VCS and their values.

Table 1-1 Attribute Data Types

Data Type	Description
string	Enclose strings, which are a sequence of characters, in double quotes ("). You do not have to enclose strings in quotes when they begin with a letter, and contains only letters, numbers, dashes (-), and underscores (_). For example: ■ A string defining a network interface such `ashme0` does not require quotes as it contains only letters and numbers. Enclosing the string in double quotes is also acceptable—"ashme0". ■ A string defining an IP address requires quotes: "100.101.102.1" because the address contains periods. A string can contain double quotes, but the quotes must be immediately preceded by a backslash. In a string, represent a backslash with two forward slashes (\\).
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: 21471183247.
boolean	A boolean is an integer with the possible values of 0 (false) and 1 (true).

Attribute dimensions

The following table summarizes the dimensions of the attributes for VCS.

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 in that list.
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{}`.

Categories of bundled agents

Bundled agents are categorized according to the type of resources they make available.

Storage agents

Storage agents make your shared disks, disk groups, volumes, and mounts highly available.

- DiskReservation (DiskRes) agent—Monitors shared disks configured using Windows Disk Management.
- Mount agent—Monitors mount paths configured using Windows Disk Management.
- Volume Manager Diskgroup (VMDg) agent—Monitors shared disks configured using Storage Foundation (SFW).
- MountV agent—Monitors mount points configured on SFW-managed shared disks.
- VVRSnap agent—Makes storage available for running fire drill in a disaster recovery environment.
- VMwareDisks agent—Enables vMotion and VMware Distributed Resource Scheduler (DRS) in VCS clusters configured and deployed on virtual machines in VMware environment.
- VMNSDg agent—Manages dynamic disk groups and mounts created on local (non-shared) and non-SCSI storage, which can work without reservation.
- NativeDisks agent—Manages the disks created on local (non-shared) and non-SCSI storage, which can work without reservation.
- NetAppSnapDrive agent—Monitors the Logical Unit Numbers (LUNs) on the NetApp Filers.
- NetAppFiler agent—Monitors the state of the NetApp Filers.
- NetApp SnapMirror agent—Monitors the replication state of filer devices.

Network agents

Network agents make your IP addresses and computer names highly available.

- NIC Agent—Monitors a NIC (Network Interface Card).
- IP Agent—Monitors an IP address.
- Lanman Agent—Monitors a virtual computer name.

FileShare agents

Fileshare agents make shared directories and subdirectories highly available.

- FileShare Agent—Monitors a shared directory.
- CompositeFileShare Agent—Monitors multiple shared directories.

Services and applications agents

Services and application agents make web sites, applications, generic services, and processes highly available.

- Internet Information Services (IIS) Agent—Monitors IIS sites and services.
- GenericService Agent—Monitors generic services.
- ServiceMonitor Agent—Monitors a service or starts a user-defined script.
- Process Agent—Monitors a process.
- MSMQ Agent—Creates a MSMQ service for a MSMQ resource, brings the MSMQ resource online, takes it offline, and monitors its status.

VCS infrastructure and support agents

The VCS infrastructure and support agents provide high availability for VCS-related operations.

- NotifierMngr Agent—Monitors a notifier process.
- Registry Replication (RegRep) Agent—Synchronizes the registries of cluster nodes by monitoring changes to the registry of the active node.
- Proxy Agent—Monitors the state of a resource on a local or remote system.
- Phantom Agent—Determines the state of service groups having resources of type None only.
- RemoteGroup Agent—Monitors and manages service groups in a remote cluster.

You can use the following agents to test VCS functionality:

- ElifNone Agent—Monitors a file.
- FileNone Agent—Monitors a file.
- FileOnOff Agent—Creates a file, monitors it, and deletes it.
- FileOnOnly Agent—Creates and monitors a file.

Which agents should I use?

The information in the table enables you to decide which agent to use depending on the resource you want to make highly available.

Table 1-3 Resources and corresponding VCS agents

Resource to Make Highly Available	Agents to Use
Shared Storage Manage disks and volumes.	In case of SFW, use VMDg and MountV. In case of LDM, use DiskRes and Mount. In case of a disaster recovery environment, use VVRSnap agent. See "VVRSnap agent" on page 36. In case vMotion and VMware Distributed Resource Scheduler (DRS) need to be enabled in VCS clusters configured and deployed on virtual machines in VMware environment, use the VMwareDisks, VMNSDg, and NativeDisks agents. See "VMwareDisks agent" on page 38. See "VMNSDg agent" on page 43. See "NativeDisks agent" on page 46. In case of a NetApp storage environment, use NetAppFiler and NetAppSnapDrive. In case you want to monitor replication state of filer devices, use NetApp SnapMirror.
IP address Adapter and the associated IP address	NIC, IP, or IPv6
Computer name (virtual name)	Lanman
File shares	Fileshare or CompositeFileShare
Sites configured using Internet Information Services (IIS)	IIS
Processes (*.pl, *.sh, *.exe, and *.bat)	Process
Windows service residing in the Service Control Manager (SCM)	GenericService or ServiceMonitor
Windows Registry	Registry Replication (RegRep)

Table 1-3 Resources and corresponding VCS agents *(continued)*

Resource to Make Highly Available	Agents to Use
Microsoft Message Queuing (MSMQ) service	MSMQ
Notifier process	NotifierMngr
State of a resource on a local or remote system	Proxy
Status of parallel service groups that do not include OnOff resources	Phantom
VCS functionality	ElifNone, FileNone,FileOnOff, and FileOnOnly
Service group in another cluster	RemoteGroup

Chapter 2

Storage agents

This chapter includes the following topics:

- About Storage agents for VCS
- DiskReservation (DiskRes) agent
- Mount agent
- Volume Manager Diskgroup (VMDg) agent
- MountV agent
- VVRSnap agent
- VMwareDisks agent
- VMNSDg agent
- NativeDisks agent
- AWS EBSVol agent
- AzureDisk agent
- GoogleDisk agent

About Storage agents for VCS

Storage agents make your shared storage highly available.

- Use the Disk Reservation and Mount agents to provide high availability for shared disks and volumes managed using Windows Logical Disk Management (LDM).

Note: The Disk Reservation and Mount agents are not supported in an SFW environment.

- Use the Volume Manager Diskgroup (VMDg) and MountV agents to provide high availability for shared disks and volumes managed using Storage Foundation.

In case of a NetApp storage environment, the Logical Unit Numbers (LUNs) on the NetApp Filer are made highly available by the NetApp Filer and NetApp SnapDrive agents instead of the VMDg and MountV agents respectively.

See "About hardware replication agents for NetApp" on page 187.

DiskReservation (DiskRes) agent

The DiskReservation agent monitors shared disks. The agent also prevents data corruption caused by network partitioning, a condition in which two cluster systems provide the same services simultaneously. This condition occurs due to network link failure and may cause data corruption. The agent ensures only one cluster system holds persistent reservation to shared disks.

During a typical network partition, one system (SystemA) has the DiskRes resource online and holds reservation to shared disks. Another system (SystemB) attempts to come online and reserve the shared disks. To avoid network partitioning, SystemB issues a bus reset and waits for the interval specified in the GracePeriod attribute to determine if SystemA is functioning. If SystemA is functioning, SystemA reacquires reservation to the shared disks. Otherwise, SystemB attempts to reserve the shared disks. The system that gains reservation to the disks is designated owner of the resource, and the resource is brought online on that system.

By default, a system is designated owner of the resource if it reserves 80 percent of the shared disks. You can modify the agent's configuration to change the ownership criteria.

Note: This agent is not supported in an SFW environment.

Agent functions

online	Reserves the shared disks each second.
offline	Releases the reservation to the shared disks.
monitor	Verifies the system holds the reservation to the shared disks.

State definitions

`ONLINE`	Indicates the system holds the reservation to the shared disks.
`OFFLINE`	Indicates the system does not hold the reservation to the shared disks.
`UNKNOWN`	Indicates the agent could not determine the status of the shared disks.

Resource type definition

```
type DiskRes (
    static i18nstr ArgList[] = { Signatures, ConfigPercentage,
    FailFastInterval, GracePeriod }
    str Signatures[]
    int ConfigPercentage = 80
    int FailFastInterval = 1
    int GracePeriod = 5
)
```

Attribute definitions

This section summarizes the required and optional attributes of the DiskRes agent.

Table 2-1 DiskRes agent required attributes

Required Attribute	Description
Signatures	An array specifying the signature of each SCSI disk. To retrieve the disk signature, use the VCS utility, "havol" with the "scsitest" option. See the *Cluster Server Administrator's Guide* for instructions on how to use the utility. Type and Dimension: string-vector

Table 2-2 DiskRes agent optional attributes

Optional Attributes	Description
ConfigPercentage	The percentage of shared disks a system must reserve to be designated owner of the DiskRes resource. When a system acquires reservation to the number of disks that equals this value, the resource is brought online on the system. Set this attribute to at least 51 percent. If you set this attribute to a value less than 50 percent, the agent changes it to 51 percent. Default is 80 percent. **Note:** If you set this attribute to a value less than 51 percent, the agent internally treats it to be 51 percent. However, this is not updated in the GUI. Type and Dimension: integer-scalar
FailFastInterval	Number of seconds the resource waits before bringing down the system after reservation fails. If during this interval the resource acquires reservation, it is not brought down. The resource is brought down in case of a reservation conflict, that is, if another system acquires reservation to the shared disks. Note that if a resource is brought down under these conditions, all service groups on the system might fail over to another system in the cluster. Default is 1 second. Type and Dimension: integer-scalar
GracePeriod	Number of seconds the agent waits before taking over disk reservation after issuing a bus reset. Default is 5 seconds. Type and Dimension: integer-scalar

Sample configuration

In the following example, the Disk Reservation agent is configured to monitor three shared disks. The resource is brought online on a system when it acquires reservation to all disks.

```
DiskRes DiskRes_A (
    Signatures = {2336572, 2336573, 2336574}
    ConfigPercentage = 100;
)
```

Note that if the shared disk is reserved by another application, the DiskReservation agent resets the disk reservation. If you use the command `hastop -all -force` to stop HAD on a system, the system holds reservation to the disk even after HAD is stopped and the disk is not visible from any other system. To see the shared disk from other systems, bring down the DiskReservation resource gracefully, using the command `hastop -all`. You could also use the havol utility to remove the reservation on the shared disks. See the description of the havol utility in the *Cluster Server Administrator's Guide* for instructions.

Mount agent

The Mount agent brings a basic disk mount point online, monitors it, and takes it offline. When a shared disk fails over to another system, the Mount agent ensures that the new system accesses the volume in the same way it was accessed before failover. The Mount agent ensures a consistent device path by mounting the volume with the same mount path (drive letter) on the new system. The agent also ensures proper dismounting from a failed system when a resource or group is taken offline. When a disk partition is mounted, VCS creates an administrative share to enable remote administration of the disk. Note that the volume to be monitored using the Mount agent must not be mounted multiple times.

The Mount agent supports basic disks only. VCS supports dynamic disks configured and controlled by SFW. To mount and unmount dynamic disks, use the MountV agent.

See "MountV agent" on page 30.

The Mount agent supports intelligent resource monitoring and uses Intelligent Monitoring Framework (IMF) for resource state change notifications. The agent uses the PnP notifications generated by the operating system. For example, PnP notifications are generated for volume arrival or departure, volume failure, and file system notifications.

For more information about IMF, refer to the *Cluster Server Administrator's Guide*.

Note: This agent is not supported in an SFW environment.

Dependency

The Mount resource depends on the DiskRes resource.

Agent functions

online	Mounts the partition (assigns a drive letter or folder) at the specified mount path.
offline	Unassigns the drive letter or folder and unmounts the partition.
monitor	Verifies that the partition is accessible from the specified mount path.

State definitions

`ONLINE`	Indicates the system can access the configured partition at the specified mount path.
`OFFLINE`	Indicates the system cannot access the configured partition at the specified path.
`UNKNOWN`	Indicates the agent could not determine the status of the resource.

Resource type definition

```
type Mount (
    static i18nstr ArgList[] = { MountPath, PartitionNo, Signature,
    ForceUnmount, ListApplications, AutoFSClean }
    static int IMF{} = { Mode=2, MonitorFreq=5, RegisterRetryLimit=3 }
    i18nstr MountPath
    int PartitionNo
    str Signature
    str ForceUnmount = READ_ONLY
    boolean ListApplications = 1
    boolean AutoFSClean = 1
)
```

Attribute definitions

This section summarizes the required and optional attributes of the Mount agent.

Table 2-3 Mount agent required attributes

Required Attributes	Description
MountPath	The drive letter or path to an empty NTFS folder that will be assigned to the partition being mounted. When configuring a directory to host the mount, verify the following conditions: ■ The configured path exists. ■ The directory is empty. ■ The volume on which the directory resides is NTFS-formatted. ■ The directory is not a system directory. The attribute can be specified as *X, X:, X:\, X:\Directory, or X:\Directory*. The options are valid only if editing the attribute from Cluster Manager (Java Console). If editing the configuration file main.cf directly, you must specify two slashes (\\) instead of one; for example, *X:\\Directory*. **Note:** This attribute can take localized values. Type-Dimension: string-scalar
PartitionNo	The partition on the disk configured for mounting. Note that the base index for the partition number is 1. Default is 0. To retrieve the partition number use the VCS utility, "havol" with the "getdrive" option. See the description of the havol utility in the *Cluster Server's Guide* for instructions. Type-Dimension: integer-scalar
Signature	A system-specified disk identifier. To retrieve the disk signature, use the VCS utility, "havol" with the "getdrive" option. See the description of the havol utility in the *Cluster Server's Guide* for instructions. Type-Dimension: string-scalar

Table 2-4 Mount agent optional attributes

Optional Attributes	Description
AutoFSClean	If the agent detects corruption, this attribute defines whether the agent automatically cleans the file system by running `Chkdsk /X` on the volume being brought online. The value 1 indicates the agent cleans the file system. The value 0 indicates it does not. Note that cleaning a file system may result in loss of data. Default is 1. Type-Dimension: boolean-scalar

Table 2-4 Mount agent optional attributes *(continued)*

Optional Attributes	Description
ForceUnmount	Defines whether the agent unmounts the volume forcibly while the volume is being used by other applications. The attribute takes the following values: ■ **NONE:** The agent does not unmount volume if other applications are accessing it. ■ **READ-ONLY:** The agent gracefully unmounts the volume even if applications are accessing it. ■ **ALL:** The agent unmounts the volume irrespective of the type of access applications have to the volume. Default is READ_ONLY. Type-Dimension: string-scalar
ListApplications	Defines whether the agent lists the applications accessing the volume while unmounting. Default is 1. Type-Dimension: boolean-scalar

Sample configurations

The following example shows a DiskRes resource with one partition.

```
DiskRes Disk_0 (
    Signatures = { 10003330 }
)

Mount MOUNT_W (
    MountPath = W
    PartitionNo = 1
    Signature = 10003330
)

Mount_W requires Disk_0
```

The next example shows two DiskRes resources, with one disk mounted from the mount point of the other disk are configured. The mount point X is created for partition 1 of Disk 0. Mount resource MOUNT_USERDATA mounts partition 1 of Disk 1 at the mount path X:\USERS\DATA.

```
DiskRes Disk_0 (
    Signatures = {10003330,398787651}
)

Mount MOUNT_X (
```

```
    MountPath = X
    PartitionNo = 1
    Signature = 10003330
)


Mount MOUNT_USERDATA (

    MountPath = "X:\\USERS\\DATA"
    PartitionNo = 1
    Signature = 398787651
    ForceUnmount = 1
    ListApplications = 1
    AutoFSClean = 1
)

MOUNT_X requires Disk_0
MOUNT_USERDATA requires MOUNT_X
```

Volume Manager Diskgroup (VMDg) agent

The Volume Manager Diskgroup agent imports, monitors, and deports a cluster disk group configured using Storage Foundation. The agent makes the disk group highly available. The Volume Manager Diskgroup agent supports Dynamic Multi-Pathing (DMP) and works in a SAN environment. The agent is represented by the VMDg resource type.

This agent supports intelligent resource monitoring and uses Intelligent Monitoring Framework (IMF) for resource state change notifications. The agent relies on the disk group related PnP notifications raised by Storage Foundation (SFW). For example, SFW raises PnP notifications for disk group import and deport state change and for disk group access state change (read-only, read/write).

For more information about IMF, refer to the *Cluster Server Administrator's Guide*.

Note: Use this agent to manage volumes on disk groups configured using Storage Foundation. Do not use this agent if you use other software to manage shared storage.

Agent functions

online	Imports the configured disk group.

offline	Deports the configured disk group.
monitor	Verifies that the configured disk group is imported.

State definitions

ONLINE	Indicates the configured disk group is imported.
OFFLINE	Indicates the configured disk group is not imported.
UNKNOWN	Indicates the agent could not detect the disk group.

Resource type definition

```
type VMDg (
static i18nstr ArgList[] = { DiskGroupName, ForceImport,
ForceDeport, ListMountedVolumes, ListApplications, VxSVCFailAction,
VxSVCRestartAttempts, DetailMonitorFreq, DGGuid, FastFailOver, ForFireDrill }
static int IMF{} = { Mode=2, MonitorFreq=5, RegisterRetryLimit=3 }
static i18nstr IMFRegList[] = { DiskGroupName, DGGuid }
str DiskGroupName
boolean ForceImport = 0
boolean ForceDeport = 0
boolean ListMountedVolumes = 0
boolean ListApplications = 0
int DetailMonitorFreq = 10
str DGGuid
str VxSVCFailAction = RESTART_VXSVC
int VxSVCRestartAttempts = 10
boolean FastFailOver = 0
boolean ForFireDrill = 0
)
```

Attribute definitions

This section summarizes the required and optional attributes for the VMDg agent.

Table 2-5 VMDg agent required attributes

Required Attribute	Description
DiskGroupName	The name of the cluster disk group configured using SFW. The disk group name may be retrieved from the SFW console by running the command `vxdg list`, or by using the VMGetDrive utility. See the description of the VMGetDrive utility in the *Cluster Server Administrator's Guide* for instructions. Type and Dimension: string-scalar

Table 2-6 VMDg agent optional attributes

Optional Attributes	Description
DetailMonitorFreq	The number of monitor cycles after which the agent monitors the resource in detail. Setting this value to 0 disables detail monitoring. Default is 10. Type and Dimension: integer-scalar
DGGuid	The disk group ID of the cluster disk group configured using SFW. The disk group GUID may be retrieved by running the command `vxdg list`, or by using the VMGetDrive utility. See the description of the VMGetDrive utility in the *Cluster Server Administrator's Guide* for instructions. Type and Dimension: string-scalar
ForceImport	Defines whether the agent forcibly imports the disk group when exactly half the disks are available. The value 1 indicates the agent imports the configured disk group when half the disks are available. The value 0 indicates it does not. Default is 0. This means that the disk group will be imported only when SFW acquires control over majority of the disks. **Note:** Set this attribute to 1 only after verifying the integrity of your data. If due caution is not exercised before setting this attribute to 1, you risk a split-brain condition, leading to potential data loss. Type and Dimension: boolean-scalar
ForceDeport	Defines whether the agent forcibly deports the disk group even if the disks within the disk group are being used. The value 1 indicates the agent forcibly deports disk groups. The value 0 indicates it does not. Default is 0. **Note:** Set this attribute to 1 only after verifying the integrity of your data. If due caution is not exercised before setting this attribute to 1, you risk a potential data loss. Type and Dimension: boolean-scalar

Table 2-6 VMDg agent optional attributes *(continued)*

Optional Attributes	Description
ListApplications	Defines whether the agent lists the applications accessing the disk group while deporting. The value 1 indicates the agent lists the applications. The value 0 indicates it does not. Default is 0. **Note:** The list of applications can be seen in the agent's log. Type and Dimension: boolean-scalar
ListMountedVolumes	Defines whether the agent lists all mounted volumes in the disk group. The value 1 indicates the agent lists the mounted volumes. The value 0 indicates it does not. Default is 0. **Note:** The list of mounted volumes can be seen in the agent's log. Type and Dimension: boolean-scalar
VxObFailAction	This attribute has been deprecated. Use VxSVCFailAction instead. Type and Dimension: string-scalar
VxObRestartAttempts	This attribute has been deprecated. Use VxSVCRestartAttempts instead. Type and Dimension: integer-scalar
VxVMFailAction	This attribute has been deprecated. Use VxSVCFailAction instead. Type and Dimension: string-scalar
VxVMRestartAttempts	This attribute has been deprecated. Use VxSVCRestartAttempts instead. Type and Dimension: integer-scalar
VxSVCFailAction	Defines the agent behavior when the Veritas Storage Agent service (VxSVC) fails. The value `RESTART_VXSVC` indicates the agent attempts to restart the VxSVC service every monitor cycle. The value `SHUTDOWN` indicates the agent attempts to restart the VxSVC service the number of times defined in the attribute VxSVCRestartAttempts. If it fails in restarting VxSVC, the agent shuts down the system. When this occurs, all service groups online on the system fail over to another system in the cluster. For example, if VxSVCRestartAttempts is set to 5, the agent attempts to start the service five times before shutting down the system. Default is `RESTART_VXSVC`. Type and Dimension: string-scalar

Table 2-6 VMDg agent optional attributes *(continued)*

Optional Attributes	Description
VxSVCRestartAttempts	The number of times the agent attempts to restart the failed VxSVC service before shutting down the system. This attribute is considered only when VxSVCFailAction is set to `SHUTDOWN`. Default is 10. Type and Dimension: integer-scalar
FastFailOver	This attribute decides whether or not a disk group is enabled for fast failover. The value 1 indicates that the agent enables fast failover for the configured disk group. The value 0 indicates that fast failover is disabled for the disk group. Default is 0. Type and Dimension: boolean-scalar
ForFireDrill	Defines whether the disk group being monitored by the agent is a fire drill disk group. The value 1 indicates that the disk group being monitored is a fire drill disk group. Default is 0, which means that the disk group being monitored is not a fire drill disk group. Type and Dimension: boolean-scalar

Sample configuration

```
VMDg DG_VCSDG (
    DiskGroupName = VCSDG
    ListMountedVolumes = 1
    DGGuid = fb69e244-930f-479e-b9f5-c54449fe2fc0
)
```

MountV agent

The MountV agent mounts, monitors, and unmounts volumes on cluster disk groups imported using the Storage Foundation. The agent supports volumes formatted using the NTFS, FAT, and FAT32 file systems.

When a cluster disk group fails over to another system, the MountV agent ensures the new system accesses the volume in the same way it was accessed before failover. The MountV agent ensures a consistent device path by mounting the disk group with the same mount point (drive letter) on the new system. The agent also mounts a volume as an NTFS folder. The agent ensures proper dismounting from a failed system when a resource or group is taken offline. When a volume is

mounted, VCS creates an administrative share to enable remote administration of the disk.

The agent performs a file system check when it brings the resource online. If a drive letter is already assigned to the volume, the agent skips the file system check, and declares the resource online. Note that the volume to be monitored using the MountV agent must not be mounted multiple times.

This agent supports intelligent resource monitoring and uses Intelligent Monitoring Framework (IMF) for resource state change notifications. The agent use the PnP notifications generated by the operating system. In addition, the agent also uses custom notifications generated by Storage Foundation (SFW). For example, PnP notifications are generated for volume arrival or departure, volume failure, and file system notifications.

For more information about IMF, refer to the *Cluster Server Administrator's Guide*.

Note: Use this agent to manage volumes on disk groups configured using Storage Foundation. Do not use this agent if you use other software to manage shared storage.

Dependency

The MountV resource depends on the VMDg resource.

Agent functions

online	Mounts the volume as a drive letter or as an NTFS folder.
offline	Unmounts the configured volume.
monitor	Verifies the volume is mounted as per the configuration.

State definitions

`ONLINE`	Indicates the configured volume is mounted and accessible.
`OFFLINE`	Indicates the mounted drive is inaccessible.
`UNKNOWN`	Indicates the agent could not determine the state of the configured volume.

Resource type definition

```
type MountV (
    static i18nstr ArgList[] = {MountPath, VolumeName,
    "VMDGResName:DiskGroupName", ForceUnmount, ListApplications,
    AutoFSClean, "VMDGResName:DGGuid", PurgeStaleMountPoints,
    CheckFSAccess, ForFireDrill, BlockMountPointAccess }
    static int IMF{} = { Mode=2, MonitorFreq=5, RegisterRetryLimit=3 }
    i18nstr MountPath
    str VolumeName
    str VMDGResName
    str ForceUnmount = READ_ONLY
    boolean ListApplications = 1
    boolean AutoFSClean = 1
    str MountResName
    boolean PurgeStaleMountPoints = 1
    boolean CheckFSAccess = 0
    boolean ForFireDrill = 0
    boolean BlockMountPointAccess = 0
    )
```

Attribute definitions

This section summarizes the required and optional attributes of the MountV agent.

Table 2-7 MountV agent required attributes

Required Attributes	Description
MountPath	The drive letter or path to an empty NTFS folder that will be assigned to the volume being mounted. When configuring a directory to host the mount, verify the following conditions: ■ The configured path exists. ■ The directory is empty. ■ The volume on which the directory resides is NTFS-formatted. ■ The directory is not a system directory. The attribute can be specified as *X, X:, X:\, X:\Directory*, or *X:\Directory*. These options are valid only if editing the attribute from Cluster Manager (Java Console). If editing the configuration file main.cf directly, you must specify two slashes (\\) instead of one; for example, *X:\\Directory*. **Note:** This attribute can take localized values. Type and Dimension: string-scalar

Table 2-7 MountV agent required attributes *(continued)*

Required Attributes	Description
VolumeName	The name of the volume to be mounted. For example, the name could be Raid1, Stripe2, Volume01, etc. Use the VMGetDrive utility to retrieve the volume name. See the description of the VMGetDrive utility in the *Cluster Server Administrator's Guide* for instructions. Type and Dimension: string-scalar
VMDGResName	The name of the Volume Manager Diskgroup (VMDg) resource on which the MountV resource depends. Type and Dimension: string-scalar

Table 2-8 MountV agent optional attributes

Optional Attributes	Description
AutoFSClean	If the agent detects corruption, this attribute defines whether the agent automatically cleans the file system by running Chkdsk /X on the volume being brought online. The value 1 indicates the agent cleans the file system. The value 0 indicates it does not. Default is 1. The output of the Chkdsk /x command is stored at `%vcs_home%\log\Chkdsk_<MountVres name>.txt`. Here, *%vcs_home%* is the default product installation directory, typically C:\Program Files\Veritas\Cluster Server. Type and Dimension: boolean-scalar

Table 2-8 MountV agent optional attributes *(continued)*

Optional Attributes	Description
ForceUnmount	Defines whether or not the agent unmounts the volume (gracefully or forcibly) when it is being used by other applications. The attribute can take the following values: ■ NONE: The agent does not unmount the volume if an application is accessing it. ■ READ_ONLY: The agent tries to enumerate the open handles in READ_ONLY mode to the configured mount points and then gracefully unmounts the mount points even if applications are accessing it in READ_ONLY mode. ■ ALL: The agent forcefully unmounts the volume irrespective of the type of access an application has. The agent first tries to obtain exclusive access to the configured mount point. The agent makes this attempt twice, with half a second delay between each unsuccessful attempt. If it fails to lock the volume even after 2 attempts, it proceeds with the unmount operation. ■ CLOSE_FORCE: The agent does not try to lock the configured mount points and proceeds directly with the forceful unmount operation. **Note:** Forceful unmount may potentially cause a data corruption. When you use ForceUnmount with CLOSE_FORCE, then before you switch or take the MountV resources offline, verify that none of the applications are accessing the configured mount points. Default is READ_ONLY. If you have enabled Intelligent Monitoring Framework (IMF) and fast failover in your cluster environment, Veritas recommends that you set this attribute value to ALL. This reduces the failover time as the agent does not require additional time to gracefully close the read-only handles on the volume. Type and Dimension: string-scalar
ListApplications	Defines whether the agent lists the applications accessing the volume while unmounting. The value 1 indicates that the agent lists the applications; the value 0 indicates that it does not. Default is 0. If you have enabled Intelligent Monitoring Framework (IMF) and fast failover in your cluster environment, Veritas recommends that you set this attribute value to 0. This reduces the failover time as the agent does not enumerate the applications that are accessing the volume. Type and Dimension: boolean-scalar

Table 2-8 MountV agent optional attributes *(continued)*

Optional Attributes	Description
PurgeStaleMountPoints	This attribute is applicable to a global cluster configuration. In case of a global fail over, this attribute defines whether the agent purges volume mount point (NTFS folder mounts) properties before bringing the folder mount points online on the remote cluster. The default value is 1, which means that the agent purges the volume mount point properties. Type and Dimension: boolean-scalar
CheckFSAccess	Defines whether or not the agent performs file system access monitoring on volumes mounted using SFW. The default value is 0 (False), which means that the file system check is disabled. **Note:** This attribute replaces the MountV agent's CheckFSAccess function. The CheckFSAccess function is deprecated. Type and Dimension: boolean-scalar
BlockMountPointAccess	This attribute defines whether the agent blocks access to the NTFS folder that is used as a folder mount point after the mount point is unmounted. For example, if C:\temp is used as a folder mount for a volume and BlockMountPointAccess attribute for the corresponding resource is set to 1, then after the mount point is unmounted, the agent blocks access to the folder C:\temp. The value 1 indicates that the folder is not accessible. The default value 0 indicates that the folder is accessible. In DR configuration, set this attribute to False if the volume is mounted on other volume which is a part of an application service group. **Note:** This attribute is applicable only for volumes mounted as folder mounts and not for drive letter mounts. Type and Dimension: boolean-scalar
ForFireDrill	Defines whether the volume being monitored by the agent belongs to the fire drill disk group. The value 1 indicates that the volume being monitored belongs to the fire drill disk group. Default is 0, which means that the volume being monitored does not belong to the fire drill disk group. Type and Dimension: boolean-scalar

Sample configuration

In the following configuration, the resource MountV_X is configured to mount the disk group VCSDG as a drive (X:). The resource MountV_X_Usrs_Data is configured to mount the disk group as a folder.

```
MountV MountV_X (
   MountPath = X
   VolumeName = Volume1
   VMDGResName = DG_VCSDG
   )

MountV MountV_X_Usrs_Data (


        MountPath = "X:\\users\\data"
   VolumeName = Raid1
   VMDGResName = DG_VCSDG
   )
VMDg DG_VCSDG
   DiskGroupName = VCSDG
   ForceDeport = 1
   ListMountedVolumes = 1
   ListApplications = 1
   )
MountV_X_Usrs_Data requires MountV_X
MountV_X requires DG_VCSDG
```

VVRSnap agent

The disaster recovery fire drill procedure tests the fault-readiness of a configuration by mimicking a failover from the primary site to the secondary site. The VVRSnap agent is designed to support a fire drill for a disaster recovery site that uses Volume Replicator as the replication method. The VVRSnap agent makes storage available for running fire drill in a disaster recovery environment.

Agent functions

Online	Detaches the mirrors from the original volumes to create point-in-time snapshots of the production data. Creates a fire drill disk group on the secondary site with a snapshot of the application data to use for testing purposes.

Offline	Joins the fire drill disk group to the application service group disk group. Snaps back the snapshot mirrors to reattach to the original volumes.

State definitions

ONLINE	Indicates that a snapshot is created and is available for fire drill.
OFFLINE	Indicates that a snapshot is not available for fire drill.
UNKNOWN	The attributes of the VVRSnap resource are invalid.
FAULTED	The VVRSnap resource goes offline unexpectedly outside of VCS control.

Resource type definition

```
type VVRSnap (
static i18nstr ArgList[] = { RVG, AppDiskGroupName, DiskGroupName }
str RVG
str AppDiskGroupName
str DiskGroupName
)
```

Attribute definitions

Table 2-9 VVRSnap agent required attributes

Required Attributes	Description
RVGName	Replicated Volume Group Name The name of the Replicated Volume Group (RVG) of the application whose snapshot is to be made available. The RVG name may be retrieved by running Volume Replicator command: `vxprint -V` Type and dimension: string-scalar
AppDiskGroupName	Application Disk Group Name The name of the SFW cluster disk group on the secondary site. It contains the data that needs to be made available for running fire drill. Type and dimension: string-scalar

Table 2-9 VVRSnap agent required attributes *(continued)*

Required Attributes	Description
DiskGroupName	Fire Drill Disk Group Name The name of the SFW cluster disk group that is split from the existing cluster disk group specified in AppDiskGroupName. Type and dimension: string-scalar

Sample configuration

```
VVRSnap FD00_VOM-FSSG-VVRSnap (
RVG = RVG_fsdg_0
AppDiskGroupName = fsdg
DiskGroupName = FD-fsdg
)
```

VMwareDisks agent

The VMwareDisks agent enables vMotion and VMware Distributed Resource Scheduler (DRS) in VCS clusters configured and deployed on virtual machines in a VMware environment.

When a VCS cluster with a shared disk is configured on virtual machines, VMware does not support VMware Distributed Resource Scheduler (DRS) and vMotion. Thus the vMotion and DRS capabilities are compromised. The solution to this issue would be to attach the disk(s) to a single virtual machine at a time in a VCS cluster. In case of a user initiated failover or a fault-induced failover, these disks would fail over (detach-attach) to the target virtual machine along with the service group. VMwareDisks agent manages the attaching and detaching of the disks to the virtual machines.

After the attach or detach operations, the tasks of initializing and managing the disks are performed by the VMNSDg agent (for SFW environment) and NativeDisks agent (for LDM environment).

Note: You must use an encrypted password in the ESXDetails attribute. This password is used to connect to the ESX/ESXi host or vCenter Server. For more information, see the Encrypting agent passwords section in the *Cluster Server Administrator's Guide*.

To ensure proper functioning of the VMwareDisks agent, verify the following:

- Ensure that the VirtualDiskMode attribute value matches the mode in which the VMware disks are attached. Otherwise, after a failover occurs, the VMwareDisks agent reverts them to the mode mentioned in the VirtualDiskMode attribute.
- The ESX/ESXi host user account has administrative privileges or is a root user. If you do not want to use the administrator user account or the root user, create a role with the required privileges for VMwareDisks resource functionality and assign this role to one or more users.
  The role assigned to the user account must have the following privileges at a minimum:
  - Low level file operations
  - Add existing disk
  - Change resource
  - Remove disk

  If the user is a vCenter user, you must assign the requisite privileges so that the user can access the datastore.

Agent functions

Online	Attaches the disks to the virtual machine.
Offline	Detaches the disks from the virtual machine.
Monitor	Verifies that the disks are attached to the virtual machine.

State definitions

ONLINE	Indicates that the disks are attached to the virtual machine.
OFFLINE	Indicates that the disks are not attached to the virtual machine.
UNKNOWN	Indicates that the agent could not detect the state of the disks.

Resource type definition

```
type VMwareDisks (
 static i18nstr ArgList[] = { ESXDetails, DiskPaths, ResourceInfo,
 IsVMHAEnabled, VMRegisterWait, VirtualDiskMode, HAInfoDetails }
 str DiskPaths{}
 str ESXDetails{}
 static int InfoInterval = 60
```

```
 boolean IsVMHAEnabled = 1
 int VMRegisterWait = 120
 str VirtualDiskMode = persistent
 str HAInfoDetails{}
)
```

Attributes

This section summarizes the required and optional attributes of the VMwareDisks agent.

Table 2-10 VMwareDisks agent required attributes

Required attribute	Description
ESXDetails	The list of host names and credentials of the ESX hosts or vCenter servers on which the virtual machines are configured. This attribute takes input in the format Key=Value. Key is the ESX or vCenter host name or IP address. Value takes the user credentials in the format User name=Encrypted password. **Note:** Do not specify the password in clear text. You must specify an encrypted password for the user. This password is used to connect to the ESX/ESXi host or vCenter server. For more information, see the Encrypting agent passwords section in the *Cluster Server Administrator's Guide*. Type and dimension: string-association
DiskPaths	The list of disk paths to be managed by the VMwareDisks resource. Specify this attribute value in the format Key=Value. Key is of the form [Data store name] Disk path. Value is of the form SCSI controller key: Target ID (unit number of the disk). For example, 0:2. For RDM, add prefix RDM: to the disk path. Alternatively, the key can be of the form Disk_UUID: [Data store name] Disk_path. In case of RDM, RDM:Disk_UUID:[Data store name] Disk_path. If the disk UUID is not provided, the agent discovers the UUID and modifies the attribute value to include UUID of the disk. Type and dimension: string-association

Table 2-11 VMwareDisks agent optional attributes

Optional attribute	Description
HAInfoDetails	Determines whether or not vSphere HA is enabled. This attribute uses the vCenter Server hostname or IP address to determine the status of vSphereHA. The value must be specified in the format: Key=Value. Where, Key= vCenter Server hostname or IP address Value= vCenter Server logon user credentials. This must be specified in the format: User name=Encrypted password. If you do not specify a value for this attribute, the agent considers the vSphereHA setting based on IsVMHAEnabled attribute value. Type and dimension: string-association
IsVMHAEnabled	Defines whether vSphere HA is enabled. The value 1 indicates that vSphere HA is enabled. This attribute value should match the vSphere HA settings in the VMware vSphere HA cluster. If vSphere HA is turned on, set this attribute value to 1 (True). If vSphere HA is turned off, set this attribute value to 0 (False). vSphereHA settings are considered based on this attribute only if the HAInfoDetails attribute is not set, or if the agent fails to retrieve the vSphere HA details based on the HAInfoDetails attribute. Default is 1 (True). Type and dimension: boolean-scalar
VMRegisterWait	The specified time interval, in seconds, during which VMware HA registers the virtual machine on any other ESX host when an ESX host fails. This is applicable only if IsVMHAEnabled is set to true. Default value is 120 seconds. Type and dimension: integer-scalar

Table 2-11 VMwareDisks agent optional attributes *(continued)*

Optional attribute	Description
VirtualDiskMode	Specifies the mode to be used when the disk is attached, and takes one of the following values: ■ `persistent` ■ `independent_persistent` ■ `independent_nonpersistent` You must modify the value after you configure application monitoring. **Note:** The VMwareDisks agent does not detect the mode in which the disk is configured. After a failover, the disk is attached in the mode that is defined in the attribute value. For details about the disk modes, refer to the VMware documentation. Default value: `persistent` Type and dimension: string-scalar

Sample configuration

Sample configuration where UUID is not provided for the DiskPaths attribute:

```
VMwareDisks VMwareDisks_1 (
 ESXDetails = { "192.168.0.100" = "root=HVJtWTwVLnINjNK",
  "192.168.0.101" = "root=HVJtWTwVLnINjNK",
  "192.168.0.102" = "root=HVJtWTwVLnINjNK" }
 DiskPaths = {
  "[SharedStorage2] VxSwapHost2_1/VxSwapHost1_1.vmdk" = "0:1",
  "[SharedStorage2] VxSwapHost2_1/VxSwapHost1_2.vmdk" = "0:2",
  "RDM:[SharedStorage2] VxSwapHost2_1/VxSwapHost1_3.vmdk" = "0:3" }
 VirtualDiskMode = independent_persistent
)
```

Sample configuration where UUID is provided for the DiskPaths attribute:

```
VMwareDisks VMwareDisks_1 (
 ESXDetails = { "192.168.0.100" = "root=HVJtWTwVLnINjNK",
  "192.168.0.101" = "root=HVJtWTwVLnINjNK",
  "192.168.0.102" = "root=HVJtWTwVLnINjNK" }
 DiskPaths = {"RDM:6000C29a-11a3-7845-029d-10737a83ced7:
  [SharedStorage2] VxSwapHost2_1/VxSwapHost1_3.vmdk" = "0:3" }
 VirtualDiskMode = independent_persistent
)
```

VMNSDg agent

The VMNSDg (Volume Manager Non-Shared Diskgroup) agent manages dynamic disk groups and mounts created on local (non-shared) and non-SCSI storage, which can work without reservation.

The agent imports, monitors, and deports a dynamic disk group configured using SFW. The agent makes the disk group highly available. The agent is represented by the VMNSDg resource type.

In combination with the VMwareDisks resource, the VMNSDg agent is responsible for:

- During import operation, ensure that the disks newly attached to the virtual machine are accessible to Windows.
- During deport operation, prepare the disks for the detach operation from the virtual machine.

Agent functions

Online	Imports the configured dynamic disk group.
Offline	Deports the configured dynamic disk group.
Monitor	Verifies that the configured dynamic disk group is imported.

State definitions

ONLINE	Indicates the configured dynamic disk group is imported.
OFFLINE	Indicates the configured dynamic disk group is not imported.
UNKNOWN	Indicates the agent could not detect the dynamic disk group state. It also indicates an error in the configuration.

Resource type definition

```
type VMNSDg (
static i18nstr IMFRegList[] = { DiskGroupName, DGGuid }
static i18nstr ArgList[] = { DiskGroupName, DGGuid,
VxSVCFailAction, VxSVCRestartAttempts, SkipStorageValidation }
str DiskGroupName
str DGGuid
str VxSVCFailAction = RESTART_VXSVC
```

```
int VxSVCRestartAttempts = 10
boolean SkipStorageValidation = 0
)
```

Attribute definitions

This section summarizes the required and optional attributes of the VMNSDg agent.

Table 2-12 VMNSDg agent required attributes

Required Attribute	Description
DiskGroupName	The name of the dynamic disk group configured using Storage Foundation (SFW). The disk group name may be retrieved from the SFW console by running the command vxdg list, or by using the VMGetDrive utility. For information on using the VMGetDrive utility, see the description of the utility in the *Cluster Server Administrator's Guide*. Type and Dimension: string-scalar
DGGuid	The disk group ID of the dynamic disk group configured using SFW. The disk group GUID may be retrieved by running the command vxdg list, or by using the VMGetDrive utility. For information on using the VMGetDrive utility, see the description of the utility in the *Cluster Server Administrator's Guide*. Type and Dimension: string-scalar

Table 2-13 VMNSDg agent optional attributes

Optional Attributes	Description
VxSVCFailAction	Defines the agent behavior when the Veritas Enterprise Administrator Service (VxSVC) fails. The value RESTART_VXSVC indicates the agent attempts to restart the VxSVC service every monitor cycle. The value SHUTDOWN indicates the agent attempts to restart the VxSVC service the number of times defined in the attribute VxSVCRestartAttempts. If the agent fails to restart VxSVC, then it restarts the system. When this occurs, all the service groups come online on another system in the cluster. For example, if VxSVCRestartAttempts is set to 5, then the agent attempts to start the service five times before restarting the system. Default is RESTART_VXSVC. Type and Dimension: string-scalar
VxSVCRestartAttempts	The number of times the agent attempts to restart the failed VxSVC service before restarting the system. This attribute is considered only when VxSVCFailAction value is set to SHUTDOWN. Default is 10. Type and Dimension: integer

Table 2-13 VMNSDg agent optional attributes *(continued)*

Optional Attributes	Description
SkipStorageValidation	Defines whether the VMNSDg agent supports the dynamic disk groups which are configured on external disks. In case of SCSI controllers, the disks are considered as internal (non-shared) if the disk port is same as the port of OS disk, otherwise the disks are considered as external disks. The external disks can be shared or non-shared. By default, this attribute blocks the disk groups configured on any external disk (shared or non-shared). Set this attribute to 1 if you want the VMNSDg agent to support disk groups configured on external SCSI disks. **Note:** Configuring a VMNSDg agent resource for a dynamic disk group created on shared storage can cause data corruption and the VMNSDg agent resource can fault after regular intervals. Veritas recommends that you use the VMDg agent if the disk groups are created on shared external disks and use the VMNSDg agent if the disk groups are created on non-shared external disks. Default value is 0. Type and Dimension: boolean-scalar

Sample configuration

```
VMNSDg D_DG (
DiskGroupName = DDG
DGGuid = fb69e244-930f-479e-b9f5-c54449fe2fc0)
```

NativeDisks agent

The NativeDisks agent manages the disks created on local (non-shared) and non-SCSI storage, which can work without reservation.

The agent brings the disks online, monitors them, and takes them offline using Logical Disk Manager for Windows. The agent makes the disks highly available.

In combination with the VMwareDisks resource, the NativeDisks agent is responsible for:

- While bringing the disks online, ensure that the disks newly attached to the virtual machine are accessible to Windows.

- While taking the disks offline, prepare the disks for the detach operation from the virtual machine.

Agent functions

Online	Brings the disks online.
Offline	Takes the disks offline.
Monitor	Verifies that the configured disks are online and accessible.

State definitions

ONLINE	Indicates the configured disks are accessible to Windows.
OFFLINE	Indicates the configured disks are not accessible to Windows
UNKNOWN	Indicates the agent could not detect the state of the disks. It also indicates an error in the configuration.

Resource type definition

```
type NativeDisks (
 static i18nstr ArgList[] = { Signatures }
 str Signatures[]
)
```

Attribute definitions

Table 2-14 NativeDisks agent required attributes

Required attribute	Description
DiskSignatures	The list of the disk signatures of all the disks that this agent monitors. Type and dimension: string-vector

Sample configuration

```
NativeDisks NativeDisks_1 (
  Signatures = { 179335408, 1601093428 }
  )
```

AWS EBSVol agent

A dedicated agent is required to provide high availability of the Amazon EBS volumes across nodes in an InfoScale cluster within the same availability zone. The EBSVol agent provides high availability of the EBS volumes during the failover of an application.

The EBSVol agent attaches EBS volumes to Amazon EC2 instances, monitors the volumes, and detaches them from the instances. It uses AWS CLI commands to determine the state of the EBS volumes.

Note: The EBS volumes should not be configured in a parallel service group.

Prerequisites

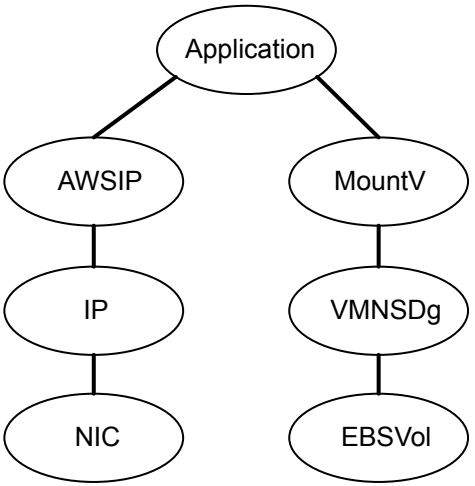
- Install the AWS CLI package.
  For details, refer to the *InfoScale Solutions in Cloud Environments* document.
- Create an IAM role with the requisite permissions, and attach the roles to the EC2 instances that form the cluster nodes.
  On the AWS Management Console, navigate to **IAM** > **Create Policy** > **Role**, and create the role.
  The following is a sample of the policy (`.json`) file:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                 "ec2:DescribeVolumes",
                 "ec2:AttachVolume",
                 "ec2:DetachVolume"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
```

Dependencies

The EBS volume resource does not depend on any other resources.

Figure 2-1 Sample service group with an EBS volume resource



Agent functions

Online — Attaches the EBS volume to an EC2 instance using the attach-volume command.

Offline — Detaches the EBS volume from the EC2 instance using the detach-volume command.

Monitor — Determines if the EBS volume is attached to the EC2 instance or not using the `describe-volume` command.

Clean — Adds the following warning in the agent log if the EBS volume is busy or cannot be detached.

```
Administrative intervention required.
```

Administrator should take corrective actions to detach the EBS volume.

State definitions

ONLINE — Indicates one of the following:

- The volume is attached to an EC2 instance or is in the Busy state on the same instance.
- The attach operation is in progress and the volume gets successfully attached to the instance within the duration that is specified in the MONITOR_TIMEOUT attribute.

OFFLINE — Indicates one of the following states:

- The detach operation is in progress and the volume gets successfully detached from the EC2 instance within the duration that is specified in the MONITOR_TIMEOUT attribute.
- The volume is not attached to any instance.
- The volume is attached to a different instance
- The volume is in the Busy state on another instance.

FAULTED — Indicates that the EBS volume has unexpectedly got detached from an EC2 instance.

UNKNOWN — Indicates that one of the following states:

- AWS CLI is not installed.
- The AWSBinDir attribute is not configured correctly.
  For example, if AWS CLI is installed at `C:\Program Files\Amazon\AWSCLI`, the AWSBinDir attribute should be set to the same value.
- The required roles are not configured or are not attached to the instance correctly.
- The ID that is specified in the VolumeID attribute is invalid.
- The volume that is specified using the VolumeID attribute belongs to a different zone.
- An attach operation is in progress, and the volume is not successfully attached to the instance within the duration that is specified in the MONITOR_TIMEOUT attribute.
- A detach operation is in progress, and the volume is not successfully detached from the instance within the duration that is specified in the MONITOR_TIMEOUT attribute.

Attributes

Table 2-15 Required attribute

Attribute	Description
VolumeId	The ID of the EBS volume. The volume and the EC2 instance must be within the same availability zone. Type and dimension: string-scalar

Table 2-16 Optional attribute

Attribute	Description
AWSBinDir	Location of the AWS EC2 commands and binaries. Type and dimension: string-scalar

Table 2-17 Internal attributes

Attribute	Description
Device	Name of the device to make the device visible to the instance. For example, `xvdf` through `xvdp`. This attribute is reserved for internal use only.
NumThreads	The number of threads that are used within the agent process for managing resources. This number does not include the number of threads that are used for other internal purposes. Setting the NumThreads attribute to a higher value may decrease the time required to go online or the time required to monitor a large number of EBS volume resources. Type and dimension: static integer-scalar Default: 1

Resource type definition

```
type EBSVol (
    static int FaultOnMonitorTimeouts = 0
    static int NumThreads = 1
    static i18nstr ArgList[] = { VolumeId, AWSBinDir, AWSDevice }
    static boolean AEPTimeout = 1
    str VolumeId
    str AWSBinDir
    temp str AWSDevice
)
```

Sample configuration

```
EBSVol res (
    VolumeId = vol-061516adc454d7e05
    AWSBinDir = "C:\Program Files\Amazon\AWSCLI"
)
```

Debug log levels

The EBSVol agent uses the DBG_1 debug log level.

AzureDisk agent

Virtual machines in Azure use data disks to store the applications data. The AzureDisk agent supports managed and unmanaged data disks and provides high availability of these disks during fail-over of an application.

The AzureDisk agent brings online, takes offline, and monitors the managed and unmanaged Azure data disks. It attaches the managed and unmanaged data disks to a virtual machine of the same or different resource group. The AzureDisk agent uses Azure Python SDK to determine whether the Azure data disks are attached to the Azure virtual machines or not.

Prerequisite

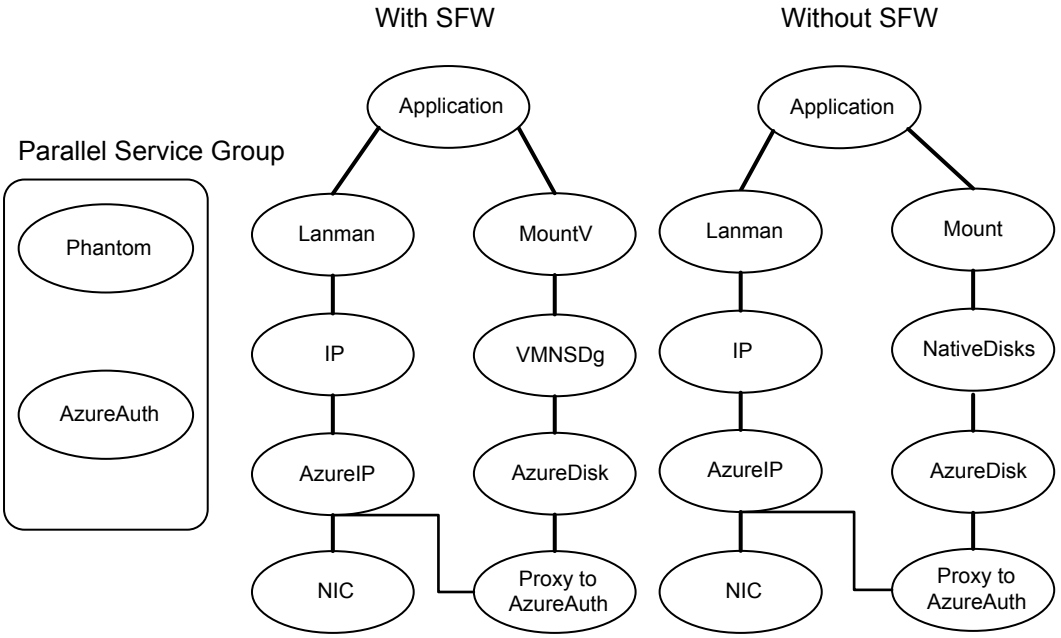
Configure the AzureAuth agent.

See “AzureAuth agent” on page 151.

Dependencies

The Azure disk resources depend on the AzureAuth resource.

Figure 2-2 Sample service group dependency



Agent functions

Open	Fetches the name of the Azure VM.
Online	Attaches Azure data disk to the Azure VM.
Offline	Detaches Azure data disk from the Azure VM.
Monitor	Determines if the Azure data disk is attached to the current Azure VM or not.
Clean	Terminates all ongoing resource actions and detaches the Azure data disk from the Azure VM.

State definitions

ONLINE	Indicates that the Azure data disk is attached to the Azure VM.
OFFLINE	Indicates that the Azure data disk is detached from the Azure VM.
FAULTED	Indicates that the Azure data disk was unexpectedly detached from the Azure VM outside of VCS control.

UNKNOWN — Indicates that a problem exists because of one of the following reasons:

- Azure Python SDK is not installed
- Azure VM is not found in the specified Resource Group
- Azure data disk is not configured correctly or has invalid ID
- Roles are not configured correctly or attached to the instance
- The disks are attached to Azure VM of another cluster
- DiskIds has a combination of managed and unmanaged disks
- Configured unmanaged disks belong to different storage accounts

Attributes

Table 2-18 Required attributes

Attribute	Description
DiskIds	In case of managed disks, resource ID of the disk. In case of unmanaged disks, VHDUri of the disk. You can add one or more disk Ids separated by a space. Type and dimension: string-vector
VMResourceGroup	Azure resource group where the Azure VM resides. Type and dimension: string-scalar
AzureAuthResName	Name of the authentication agent resource that handles Azure related authentication. Type and dimension: string-scalar
StorageAccountKey	Access key of the storage account. You must encrypt this key by using the vcsencrypt utility. For details, see the *Cluster Server Administrator's Guide*. This attribute is required only if unmanaged disks are configured. Type and dimension: string-scalar

Table 2-19 Optional attribute

Attribute	Description
AzureVMName	Name of the VM in Azure on which agent is running. Type and dimension: string-scalar

Resource type definition

```
type AzureDisk (
    static int MonitorTimeout = 120
    static i18nstr ArgList[] = { StorageAccountKey, tempVMName,
      DiskIds, VMResourceGroup, AzureVMName,
      "AzureAuthResName:SubscriptionId","AzureAuthResName:ClientId",
      "AzureAuthResName:SecretKey", "AzureAuthResName:TenantId" }
    str DiskIds[]
    str VMResourceGroup
    str AzureVMName
    str AzureAuthResName
    temp str tempVMName
    str StorageAccountKey
)
```

Sample configuration

Sample configuration with managed disks:

```
AzureDisk azure-res-disk (
    DiskID = "/subscriptions/6940a326-fgh6-40dd-b616-c1e9bbdf1d63/
      resourceGroups/azureRG/providers/Microsoft.Compute/disks/
      AzureManagedDisk"
    VMResourceGroup = "azureVMRG"
    AzureAuthResName = "azure-auth-res"
)
```

Sample configuration with unmanaged disks:

```
AzureDisk azure-res-disk (
    DiskIds = "https://azureunstorageaccount2.blob.core.windows.net/
      vhds/unmanageddisk.vhd"
    VMResourceGroup = "azureVMRG"
    AzureAuthResName = "azure-auth-res"
    storageAccountKey = fpjNjrNrgRupNnnLhlKHnlRjpKLitIpjTgnTpfNglSls
)
```

Debug log levels

The AzureDisk agent uses the DBG_1 debug log level.

GoogleDisk agent

The GoogleDisk agent lets you configure data disks for HA in GCP, so that the disks can also be failed over during an application failover between instances.

To store the data of the applications that you configure for HA, InfoScale supports the following storage types with GCP VM instance:

- Zonal persistent disks—Standard and SSD
- Regional persistent disks—Standard and SSD

The GoogleDisk agent brings the GCP data disks online, monitors their status, and takes them offline. The GoogleDisk agent uses GCP Python SDK to determine whether the data disks are attached to the virtual machines or not.

Prerequisites

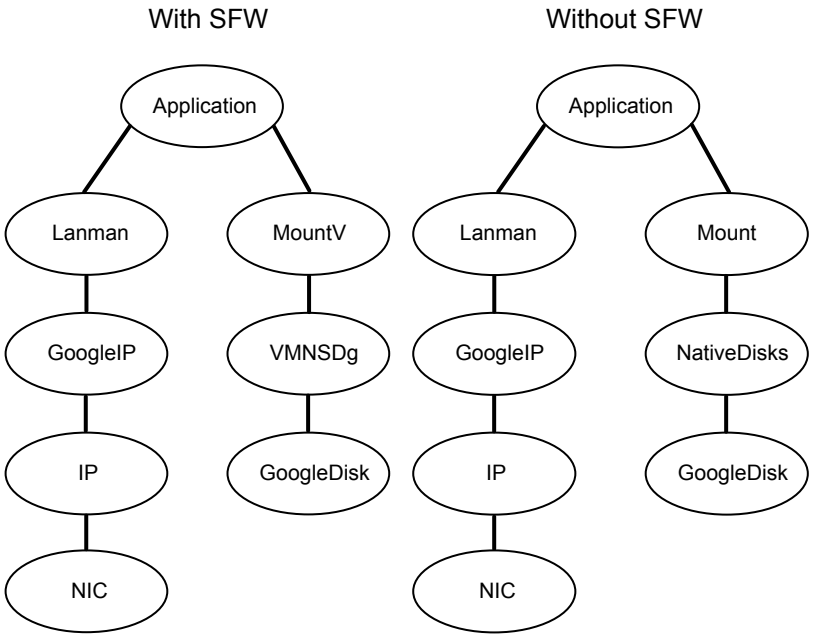
- Your Google service accounts must have the following permissions, at the least:
  - `compute.projects.get` (at the project level, to fetch the project metadata)
  - `compute.instances.get` (at the instance level, to fetch the instance metadata)
  - `compute.instances.attachDisk` (at the instance level, to attach a disk to the instance)
  - `compute.instances.detachDisk` (at the instance level, to detach a disk from the instance)
  - `compute.zoneOperations.get` (to check the status of operations in the zone)
  - `compute.disks.get` (to fetch the list of the current users of a disk)
- To access Google APIs, the following Python modules must be present on the InfoScale cluster nodes:
  - `oauth2client`
  - `google-api-python-client`
- You must first create a disk before you configure a resource for it. Otherwise, the online operation fails.
- Add dependencies in the service group as depicted in Figure 2-3

Install Python SDK for GCP on each of the cluster nodes by running the following commands in `C:\Program Files\Veritas\VRTSPython`, or wherever the `VRTSPython` folder is installed on the node:

- `python -m pip install --upgrade pip`

- `python -m pip install --upgrade google-api-python-client`
- `python -m pip install --upgrade oauth2client`

Figure 2-3 Sample service group dependency



Agent functions

Open	Fetches the current GCP VM instance name and the zone and project to which the instance belongs, using its metadata.
Online	Attaches the GCP disk to the VM instance.
Offline	Detaches the GCP disk from the VM instance.
Monitor	Determines whether the GCP data disk is attached to the current VM instance or to any other VM instance within or outside the cluster. It also identifies the mode in which the disk is attached and accordingly returns a status value.
Clean	Detaches the GCP disk from the VM instance.

State definitions

ONLINE — Indicates that the GCP disk is attached to the current VM instance.

OFFLINE — Indicates one of the following:

- The GCP disk is not in use by any VM instance.
- The disk is detached from the current VM instance, but it is attached to another VM instance within the cluster in the read-write mode.
- The disk is attached to any other VM instance within or outside the cluster in the read-only mode

FAULTED — Indicates that the GCP disk was unexpectedly detached from the VM instance outside of VCS control.

UNKNOWN — Returns UNKNOWN if:

- The required Google Python SDKs are not installed on the current VM instance.
- It fails to fetch the required attributes values.
- The value of the ReadOnly attribute does not match the mode in which the GCP disk is attached to the VM instance. Such a situation may cause issues for the applications that use the disk.
- Roles are not configured correctly or they are not correctly associated with the VM instance so as to fetch the disk information.
- An invalid DiskName is provided.
- The specified DiskName of a zonal disk is not found in the zone in which the VM instance exists. The zonal disk exists in some other zone.
- The disk is attached to a VM instance outside the cluster in the read-write mode.

Attributes

Table 2-20 Required attributes

Attribute	Description
DiskName	Name of the disk. Type and dimension: string-scalar
ReadOnly	Mode in which to attach the disk: ■ 0: READ_WRITE ■ 1: READ_ONLY Default value: 0 (READ_WRITE) Type and dimension: boolean-scalar

Table 2-20 Required attributes *(continued)*

Attribute	Description
RegionalDisk	Type of the disk: ■ 0: zonal disk ■ 1: regional disk **Note:** Regional disks can fail over across zones, and they can be attached in the READ_WRITE mode only. Default value: 0 (zonal disk) Type and dimension: boolean-scalar

Table 2-21 Optional attribute

Attribute	Description
ProjectID	Project to which the VM instance belongs. Type and dimension: string-scalar
Zone	Zone to which the instance belongs. Type and dimension: string-scalar
VMName	Name of the GCP VM instance on which the agent runs. Type and dimension: string-scalar

Resource type definition

```
type GoogleDisk (
    static int FaultOnMonitorTimeouts = 0
    static str ArgList[] = { DiskName, ReadOnly,
      RegionalDisk, ProjectID, Zone, VMName,
      tempProjectName, tempVMName, tempZone }
    boolean ReadOnly = 0
    boolean RegionalDisk = 0
    str DiskName
    str ProjectID
    str Zone
    str VMName
    temp str tempProjectName
    temp str tempZone
    temp str tempVMName
)
```

Sample configuration

```
GoogleDisk VVRSG-GoogleDisk (
    DiskName = sql-disk
)
```

Debug log levels

The GoogleDisk agent uses the DBG_1 and the DBG_2 debug log levels.

Chapter 3

Network agents

This chapter includes the following topics:

- About network agents
- NIC agent
- IP agent
- IPv6 agent
- Lanman agent
- AWSIP agent
- AzureIP agent
- AzureDNSZone agent
- GoogleIP agent

About network agents

Network agents make IP addresses and computer names highly available.

- The NIC, IP, and IPv6 agents work together to make a virtual IP address highly available.
- The Lanman agent makes a virtual computer name highly available. The Lanman agent requires the IP agent for operation.

NIC agent

This agent monitors the configured network interface card (NIC) resource. Some NICs maintain their connection status in a system-defined variable. The NIC agent

uses this variable to determine the status of the NIC resource. If the NIC does not maintain its connection status, the agent uses a ping or a User Datagram Protocol (UDP) echo broadcast to determine the status of the resource.

This agent supports pure IPv4, pure IPv6, and mixed mode (IPv4 and IPv6) configurations.

This agent supports intelligent resource monitoring and uses Intelligent Monitoring Framework (IMF) for resource state change notifications. The agent relies on the network and hardware events raised by the operating system. For example, an event is raised when a network adapter is disabled.

For more information about IMF, see the *Cluster Server Administrator's Guide*.

Agent functions

online	Not applicable.
offline	Not applicable.
monitor	If the NIC maintains its connection status, the agent reads the system-defined variable to determine the status of the resource. If the NIC does not maintain its connection status, the agent verifies that the NIC has been assigned an IP address, and sends a ping to all the hosts listed in the attribute PingHostList. The agent counts the number of packets received by the NIC before and after the ping. If the count increases, the resource is marked `ONLINE`. If the count remains unchanged, the agent broadcasts a UDP echo request to the address derived from the administrative IP address and the associated subnet mask. The agent counts the number of packets received by the NIC before and after the broadcast. If the count increases, the resource is marked `ONLINE`. If the count remains the same or decreases, the resource is marked `FAULTED`. The NIC agent also calculates the percentage of erroneous packets transmitted during each monitor interval. If this percentage exceeds the value in the MaxTxErrorPercentage attribute for the number of consecutive monitor intervals specified by the value in the MaxTxErrInterval attribute, the NIC resource is marked `FAULTED`.

State definitions

`ONLINE`	Indicates the NIC resource is online.
`FAULTED`	Indicates a problem with the NIC or that a network link failed.
`UNKNOWN`	Indicates the agent encountered errors while monitoring the NIC resource.

Resource type definition

```
type NIC (
    static i18nstr ArgList[] = {MACAddress, PingTimeoutMseconds,
    MaxTxErrorPercentage, MaxTxErrInterval, UseConnectionStatus,
    PingHostList }
    static int IMF{} = { Mode=3, MonitorFreq=5, RegisterRetryLimit=3 }
    str MACAddress
    int PingTimeoutMseconds = 1000
    int MaxTxErrorPercentage = 50
    int MaxTxErrInterval = 10
    boolean UseConnectionStatus = True
    str PingHostList[]
)
```

Attribute definitions

Table 3-1 NIC agent required attribute

Attribute	Description
MACAddress	The physical (MAC) address, interface name, or GUID of the NIC to be monitored. This attribute is always local, that is, it is different for each system. When you use the service group configuration wizard, it sets this attribute to the MAC address by default. To monitor the interface based on its name or GUID, you must edit this attribute after you configure the service group. If NIC teaming is configured and the same interface name is used on all the cluster nodes, you might want to set this attribute to the interface name, because it remains the same for the cluster. If you use the interface name, you must specify either the appropriate name for the local node or the common interface name in case of NIC teaming, if applicable. Use the VCS Java Console to edit this attribute and enter the interface name or GUID instead of the MAC address. If you edit this attribute in a configuration file and if the interface name includes spaces, you must enter the value in double quotes. For details on modifying the attribute value, see the *Cluster Server Administrator's Guide*. **Note:** If the interface name or the GUID changes, you must update the attribute value accordingly. Type and Dimension: string-scalar

Table 3-2 NIC agent optional attributes

Attribute	Description
MaxTxErrInterval	The number of consecutive monitor intervals for which the percentage of erroneous packets should not exceed the value specified in MaxTxErrorPercentage. When this occurs, the NIC is declared FAULTED. Default is 10. Type and Dimension: integer-scalar
MaxTxErrorPercentage	The threshold error percentage for the total packets sent between two consecutive monitor intervals. Default is 50. Type and Dimension: integer-scalar
PingHostList	A list of hosts that are always reachable from the NIC. If the UseConnectionStatus attribute is set to 0, make sure the attribute PingHostList is not empty. Type and Dimension: string-vector
PingTimeoutMseconds	The timeout interval, in milliseconds, for a ping. Default is 1000 milliseconds. Type and Dimension: integer-scalar
UseConnectionStatus	Defines whether the NIC maintains its connection status. The value True indicates the NIC maintains its status. The value False indicates it does not. If this attribute is set to False, you must specify a value for the attribute PingHostList. Default is True. VCS provides a utility, "NICTest," which determines whether a NIC maintains its connection status. You can use the NICTest utility to determine whether or not this attribute is required. See the description of the NICTest utility in the *Cluster Server Administrator's Guide* for instructions. Type and Dimension: boolean-scalar

Sample configuration

In the following example, the NIC agent is configured to monitor the adapter with MAC address 00-B0-D0-31-7D-FA. The UseConnectionStatus attribute is set to False. The agent sends a ping to the hosts listed in the PingHostList attribute to determine the status of the NIC resource. If the percentage of erroneous packets transmitted exceeds 30 percent for three consecutive monitoring intervals, the NIC resource is marked FAULTED.

```
NIC NIC_RL20001 (
    MACAddress = "00-B0-D0-31-7D-FA"
    PingTimeOutMseconds = 500
    MaxTxErrorPercentage = 30
    MaxTxErrInterval = 3
    UseConnectionStatus = False
    PingHostList = { "172.16.1.19", "172.16.1.21" }
)
```

IP agent

The IP agent assigns a virtual IP address to the network interface card (NIC), monitors the IP address, and removes it.

This agent supports intelligent resource monitoring and uses Intelligent Monitoring Framework (IMF) for resource state change notifications. The agent relies on the network and hardware events raised by the operating system. For example, an event is raised when an IP address becomes unavailable.

For more information about IMF, refer to the *Cluster Server Administrator's Guide*.

Dependency

The IP resource depends on the NIC resource.

Agent functions

online	Assigns a unique virtual IP address to an empty slot on the NIC after verifying that the IP address is not being used by another system.
offline	Removes the virtual IP address assigned to the NIC.
monitor	Verifies that the specified virtual IP address is assigned to the NIC.

State definitions

`ONLINE`	Indicates the specified virtual IP address is assigned to the NIC.
`OFFLINE`	Indicates the specified virtual IP address is not assigned to the NIC.
`UNKNOWN`	Indicates the agent encountered errors while monitoring the IP resource. This may be due to an invalid or incorrect configuration.

`FAULTED` Indicates that the agent was unable to bring the IP address online. This may be because the IP address was removed out of VCS control, or the IP address is already online on the network.

Resource type definitions

```
type IP (
    static i18nstr ArgList[] = {Address, SubNetMask, MACAddress}
    static int IMF{} = { Mode=3, MonitorFreq=5, RegisterRetryLimit=3 }
    str Address
    str SubNetMask
    str MACAddress
)
```

Attribute definitions

Table 3-3 IP agent required attributes

Attribute	Description
Address	A unique virtual IP address to be assigned to the NIC. Type and Dimension: string-scalar
SubNetMask	The subnet mask associated with the IP address. Type and Dimension: string-scalar

Table 3-3 IP agent required attributes *(continued)*

Attribute	Description
MACAddress	The physical (MAC) address, interface name, or GUID of the NIC to be monitored. This attribute is always local, that is, it is different for each system. When you use the service group configuration wizard, it sets this attribute to the MAC address by default. To monitor the interface based on its name or GUID, you must edit this attribute after you configure the service group. If NIC teaming is configured and the same interface name is used on all the cluster nodes, you might want to set this attribute to the interface name, because it remains the same for the cluster. If you use the interface name, you must specify either the appropriate name for the local node or the common interface name in case of NIC teaming, if applicable. Use the VCS Java Console to edit this attribute and enter the interface name or GUID instead of the MAC address. If you edit this attribute in a configuration file and if the interface name includes spaces, you must enter the value in double quotes. For details on modifying the attribute value, see the *Cluster Server Administrator's Guide*. **Note:** If the interface name or the GUID changes, you must update the attribute value accordingly. Type and Dimension: string-scalar

Sample configuration

```
IP IP_192_168_100_20 (
    Address = "192.168.100.20"
    SubNetMask = "255.255.255.0"
MACAddress @SYSTEM1 = "00-60-08-08-BA-C4"
MACAddress @SYSTEM2 = "00-90-27-28-5A-7E"
)
```

IPv6 agent

The IPv6 agent is designed specifically to support Internet Protocol version 6 (IPv6). The agent assigns a virtual IPv6 address to the network interface card (NIC), monitors the IPv6 address, and removes it. The agent also monitors the associated IPv6 prefix on the NIC.

You can use this agent when you want to monitor a single IPv6 address on a single adapter. The IPv6 setting on the interface must be enabled with a physical (or administrative) base IPv6 address before you can assign it a virtual IPv6 address.

The IPv6 agent currently supports the following types of IPv6 addresses:

- Unicast addresses
  Only Global Unicast and Unique Local Unicast addresses are supported.
- Automatic configuration
  Only Stateless IPv6 address configuration is supported. Stateless addresses are based on the information received from the routers configured on the network.

Dependency

The IPv6 resource depends on the NIC resource.

Agent functions

online	Retrieves the IPv6 address from the configured attributes, verifies that the IPv6 address is unique and not in use already and then assigns the address to the NIC.
monitor	Retrieves the IPv6 address from the configured resource and verifies that the IPv6 address is assigned to the NIC.
offline	Removes the IPv6 address assigned to the NIC.
clean	Removes the IPv6 address assigned to the NIC.

State definitions

`ONLINE`	Indicates the specified virtual IPv6 address is assigned to the NIC.
`OFFLINE`	Indicates the specified virtual IPv6 address is not assigned to the NIC.
`UNKNOWN`	Indicates the agent encountered errors while monitoring the IPv6 resource. This may be due to an invalid or incorrect configuration, or an unsupported IPv6 address type.
`FAULTED`	Indicates that the agent was unable to bring the IPv6 address online. This may be because the IPv6 address was removed out of VCS control, or the IPv6 address is already online on the network.

Resource type definition

```
type IPv6 (
    static i18nstr ArgList[] = { Address, Prefix, MACAddress }
    str Address
    str Prefix
    str MACAddress
)
```

Attribute definitions

Table 3-4 IPv6 agent required attributes

Required Attributes	Description
Address	A unique virtual IPv6 address to be assigned to the NIC. **Note:** If you configure the IPv6 resource manually, then the IPv6 address that you specify for this attribute must not be a physical (or administrative) base IP of the cluster node. Type and Dimension: string-scalar
Prefix	The prefix associated with the IPv6 address. The Prefix is generally represented as: *ipv6-address/prefix-length* The ipv6-address variable represents the IPv6 address that follows general IPv6 addressing rules (refer to RFC 2373 for details). The prefix-length variable is a decimal value that indicates the number of leftmost contiguous higher order bits of the address that make up the network portion of the address. Example: 2001:db8:0:1::/64 Type and Dimension: string-scalar
MACAddress	The physical address of the NIC to which the virtual IPv6 address is to be assigned. Note that this attribute is always local, that is, it is different for each system. **Note:** Use the `ipconfig -all` command to retrieve the physical address of a NIC. Type and Dimension: string-scalar

Sample configuration

Here's a sample configuration of an IPv6 resource:

```
IPv6 IPRes (
       Address = "2001:db8:0:1:d78:341e:2f7c:5879"
       Prefix = "2001:db8:0:1::/64"
       MACAddress @SYSTEM1= 00-50-56-9F-53-DE
       MACAddress @SYSTEM2 = 00-10-18-10-CB-1F
       MACAddress @SYSTEM3 = 00-10-18-10-CA-DC
       )
```

Lanman agent

The Lanman agent enables clients to access data and applications on a cluster node by associating the virtual IP address with the virtual computer name in the WINS database. The agent provides the option of associating multiple IP addresses from different subnets with the virtual computer name. The agent also provides the option of creating the virtual computer object in any organizational unit in the Active Directory and enhances the DNS updating capabilities of VCS.

The Lanman agent registers the following services with the WINS server:

- Server (20h)
- Workstation (00h)
- Messenger (03h)

In an IPv6 environment, the Lanman agent relies on the DNS records to check for duplicate server names on the network. If the DNS entry for a virtual server name that uses an IPv6 address does not exist, the agent fails to validate the server for name duplication on the network. If the virtual IP is an IPv6 address, you must specify the DNS server IP address, either in the network adapter properties or in the Lanman agent's AdditionalDNSServers attribute.

The agent supports Kerberos authentication by providing the option of adding the virtual computer name to the Active Directory and adding the virtual IP address to the DNS. The agent uses the VCS Helper service user context for AD and DNS updates.

The Lanman agent updates and monitors the canonical name (CNAME) mapping in the domain name server when failing over applications across subnets (performing a wide-area failover.) The Lanman agent also supports creating DNS records in different DNS zones.

If security policies are enabled, ensure that the startup type of the Server Service is set to Automatic.

Dependency

The Lanman resource depends on the IP (in case of IPv4) and IPv6 (in case of IPv6) resource. In case of AWSIP resource, Lanman resource depends on AWSIP.

If you change your Lanman resource dependency to a new IP resource and bring the Lanman resource online, a ping to the virtual name might respond from the IP address of the previous IP resource until the next WINS broadcast. The WINS broadcast updates the WINS database with the changed association.

For example, if you took the Lanman resource offline, changed the Lanman resource dependency from IP_A to IP_B, and brought the Lanman resource online, a ping to the virtual name might still respond from IP_A. Note that the IP_A resource is kept online during this process. The ping will respond from IP_B after the next WINS broadcast updates the WINS database.

Agent functions

Online	Binds the IP addresses with the specified virtual computer name. The agent also queries the name server of the domain for Host (A), PTR, and CNAME records and adds or updates the records on the name server .
Offline	Removes the IP address binding from the virtual computer name.
Monitor	Verifies the IP addresses are bound to the virtual computer name. If DNSUpdateRequired is enabled and the resource is online, then the Lanman agent refreshes the resource records on the DNS servers. The agent queries the name servers for DNS records. It reports back `ONLINE` if the response from all the name servers contains the Host (A), PTR, and CNAME records. If no servers return the appropriate records, the monitor reports the resource as `OFFLINE`.

State definitions

`ONLINE`	Indicates the IP addresses are bound to the virtual computer name and the DNS records are as expected.
`OFFLINE`	Indicates the IP addresses are not bound to the virtual computer name or the agent failed to create the DNS records or the expected DNS records were not found.
`UNKNOWN`	Indicates the agent could not determine the status of the resource.

Resource type definition

```
type Lanman (
    static keylist LogDbg = {DBG_21}
    static i18nstr ArgList[] = {VirtualName,
    MultiNet, "IPResName:Address", "IPResName:SubNetMask",
    "IPResName:MACAddress", MultiNetInfo, DNSUpdateRequired,
    ADUpdateRequired, DNSCriticalForOnline,
    ADCriticalForOnline, ADContainer, DNSOptions,
    AdditionalDNSServers, DNSRefreshInterval, DNSZones, AliasName,
    TSIGKeyFile, TTL}
    str VirtualName
    str IPResName
    boolean MultiNet = 0
    str MultiNetInfo[]
    boolean DNSUpdateRequired = 0
    boolean ADUpdateRequired = 0
    boolean DNSCriticalForOnline = 0
    boolean ADCriticalForOnline = 0
    str ADContainer
    str DNSOptions[]
    str AdditionalDNSServers{}
    int DNSRefreshInterval = 0
    str DNSZones{}
    str AliasName = ""
    str TSIGKeyFile = ""
    int TTL = 0
)
```

Attribute definitions

Table 3-5 Lanman agent required attributes

Required Attributes	Description
IPResName	The name of the IP resource on which the Lanman resource depends. Do not define a value for this attribute if the MultiNet attribute is set to 1. Type and Dimension: string-scalar

Table 3-5 Lanman agent required attributes *(continued)*

Required Attributes	Description
VirtualName	The virtual computer name to be assigned to the server. The virtual name must be fewer than 15 characters. Note that if you specify a virtual computer name in lowercase letters, the agent converts it to uppercase. For example, the name VCSServer is converted to VCSSERVER. Type and Dimension: string-scalar

Table 3-6 Lanman agent optional attributes

Optional Attributes	Description
ADContainer	Specifies the distinguished name of the Active Directory container or the organizational unit (OU) for the newly created computer object. If no value is specified for this attribute, the Lanman resource creates the computer object in the default container "Computers." Note that the user account for VCS Helper service must have adequate privileges on the specified container to create and update computer accounts. Refer to Microsoft documentation for information on assigning user privileges for a container. By default, the attribute contains no value. **Note:** Value specified for this attribute will be effective only if ADUpdateRequired is set to 1. Type and Dimension: string-scalar
ADCriticalForOnline	Defines whether the Lanman resource faults if the agent fails to update the Active Directory. The value 1 indicates that the resource faults in case of a failure to update the Active Directory. The value 0 indicates that it does not. Default is 0. Type and Dimension: boolean-scalar

Table 3-6 Lanman agent optional attributes *(continued)*

Optional Attributes	Description
AdditionalDNSServers	An array that specifies the IP addresses of the additional DNS servers that will be updated by the Lanman resource. For all the Windows DNS servers, the forward and reverse lookup zones must be configured. For all the Berkeley Internet Name Domain (BIND) servers, only the forward lookup zones are required. All additional DNS servers are considered as Windows DNS servers by default. If any additional DNS server is a Berkeley Internet Name Domain (BIND) server, you will have to specify it in the attribute value. Example: "{"10.212.108.9" = "","10.212.108.10" = "BIND"}" Where 10.212.108.9 is the IP address of a Windows DNS server, and 10.212.108.10 is the IP address of a BIND DNS server. By default, the attribute contains no value. Values specified for this attribute will be effective only if DNSUpdateRequired is set to 1. The Lanman agent creates only CNAME records on BIND servers. You must also specify the AliasName attribute in case of BIND server updates. **Note:** The Lanman agent supports BIND version 8 and above. **Note:** In cases where the default DNS is a BIND DNS server, set the value of the DNSOptions attribute to IgnoreDefault, and specify the BIND DNS server details in this attribute. **Note:** If the BIND DNS servers are configured for secure updates, then you must configure the TSIG keys either in the DNSZones attribute or the TSIGKeyFile attribute. Type and Dimension: string-association
ADUpdateRequired	Defines whether the Lanman resource updates the Active Directory with the virtual name. The value 1 indicates that the agent updates the Active Directory. The value 0 indicates it does not. Default is 0. Type and Dimension: boolean-scalar

Table 3-6 Lanman agent optional attributes *(continued)*

Optional Attributes	Description
DNSCriticalForOnline	Defines whether the Lanman resource faults if the agent fails to update the DNS. The value 1 indicates that the resource faults in case of a failure to update the DNS. The value 0 indicates that it does not. Default is 0. In case of an IPv6 environment this attribute has no effect on the Lanman agent behavior. Even if this attribute is set to 0, the Lanman resource faults if the agent is unable to update the DNS. Type and Dimension: boolean-scalar
DNSOptions	An array that specifies the way in which the Lanman resource updates the DNS servers. This attribute can take one or all of the following values: ■ UpdateAll: Updates all the default DNS servers specified in the TCP/IP properties for the cluster node, and the additional DNS servers specified in the AdditionalDNSServers attribute. ■ IgnoreDefault: Ignores the default DNS servers and updates only the additional DNS servers. ■ PurgeDuplicate: Removes duplicate DNS entries from the DNS servers. Veritas recommends you set this value for service groups configured for wide area failover. ■ SkipPtrRecords: The Lanman resource excludes the PTR records while updating the resource records on the specified DNS servers. Any combination of these values can be specified for the attribute. This attribute takes effect only when the Lanman resource comes online. See Updating DNS servers for more information. By default, the attribute contains no value. Values specified for this attribute will be effective only if DNSUpdateRequired is set to 1 and additional DNS servers are specified in the AdditionalDNSServers attribute. **Note:** In cases where the default DNS is a BIND DNS server, set this attribute value to IgnoreDefault, and specify the BIND DNS server details in the AdditionalDNSServers attribute. Type and Dimension: string-vector

Table 3-6 Lanman agent optional attributes *(continued)*

Optional Attributes	Description
DNSUpdateRequired	Defines whether the Lanman resource updates the DNS with the virtual IP address. The value 1 indicates that the resource updates the DNS. The value 0 indicates it does not. Default is 0. If you set this attribute but there are no DNS servers specified in the TCP/IP properties, then you must specify the DNS servers that you wish to update in the AdditionalDNSServers attribute. If NetBios is disabled over TCP, set this attribute value to 1. In case of an IPv6 environment this attribute has no effect on the Lanman agent behavior. Even if this attribute is set to 0, the Lanman agent updates the DNS with the virtual IPv6 address entry by default. **Note:** If the virtual IP is an IPv6 address, you must specify the DNS server IP address, either in the network adapter properties or in the Lanman agent's AdditionalDNSServers attribute. **Note:** The Lanman resource does not update the DNS for the manual DNS entries. See Updating manual DNS entries for more information. Type and Dimension: boolean-scalar
DNSRefreshInterval	This attribute represents the time interval, in seconds, after which the Lanman agent attempts to refresh the resource records (RRs) on the DNS servers. You must set a value for this attribute if you want the Lanman agent to refresh the records on the DNS servers. The default value zero indicates that the Lanman agent does not attempt to refresh the records on the DNS servers. The Lanman agent logs warning messages if it is not able to refresh the DNS records. After 3 failed attempts, the Lanman agent logs error messages for subsequent failures. If the Lanman agent is unable to refresh the DNS records, and the records are removed as a result of a scavenging operation or by the DNS administrator, the Lanman resource will fault. Type and Dimension: integer-scalar

Table 3-6 Lanman agent optional attributes *(continued)*

Optional Attributes	Description
DNSZones	An array that specifies a list of DNS zones (in case of multi-domain environments with parent-child configurations) for which the Lanman resource should create and update Address (A) records and canonical name (CNAME) records in the DNS server of the parent domain. Example: {"child1.company.com", "child2.company.com"} Where child1.company.com and child2.company.com are DNS zones representing different child domains. By default, the attribute contains no value. This means that the Lanman agent will create and update resource records only in the DNS name servers for the zones in which the nodes exist. If multiple zones are being updated on BIND DNS servers that are configured for secure updates, then each zone may require a different TSIG key. In such a case, you must specify the absolute path of the TSIG key file in the attribute value. Example: {"child1.company.com" = "C:\TSIGKey1.key", "child2.company.com" = "C:\TSIGKey2.key"} Where TSIGKey1.key is the TSIG key for the DNS zone child1.company.com, and TSIGKey2.key is the TSIG key for the DNS zone child2.company.com. **Note:** The Lanman agent supports BIND version 8 and above. Type and Dimension: string-association
AliasName	A string representing the alias to the canonical name. The Lanman agent creates a CNAME record using the value specified in this attribute. Example: "www" Where www is the alias to the canonical name mtv.veritas.com. By default, the attribute contains no value. **Note:** This attribute is required if a BIND DNS server is specified in the AdditionalDNSServers attribute. Type and Dimension: string-scalar

Table 3-6 Lanman agent optional attributes *(continued)*

Optional Attributes	Description
TSIGKeyFile	Required when you configure BIND DNS for secure updates. Specify the absolute path to the file that contains the private Transaction Signature (TSIG) key. This key is used by the nsupdate utility to perform secure BIND DNS updates. See the BIND man pages for more information about secure DNS updates. You must copy the files containing the keys (typically the .key and the .private file) on each of the nodes that is listed in the service group's SystemList. By default, the attribute contains no value. Example: C:\TSIG\Kveritas.com.+157+00000.key **Note:** The Lanman agent supports BIND version 8 and above. Type and Dimension: string-scalar
TTL	This value defines the Time To Live (TTL) value (in seconds) that gets stored in the DNS records created by the agent. Default: 0 Example: TTL = 7200 Type and Dimension: integer-scalar
MultiNet	Defines whether the Lanman resource binds multiple IP addresses with the virtual name. The value 1 indicates the resource binds multiple IP addresses specified in MultiNetInfo with the virtual computer name. The value 0 indicates the resource binds a single IP address specified in IPResName. Default is 0. Type and Dimension: boolean-scalar

Table 3-6 Lanman agent optional attributes *(continued)*

Optional Attributes	Description
MultiNetInfo	An array that specifies details of the IP addresses to be bound to the virtual computer name. If MultiNet is set to 1, configure this attribute in the following format: MultiNetInfo = { "IP=*ip_address1* Mask=*subnetmask1* WINS=*wins_ip_address1* MACAddress=*macaddress1*", "IP=*ip_address2* Mask=*subnetmask2* WINS=*wins_ip_address2* MACAddress=*macaddress2*" } **Note:** Specifying Mask and MACAddress is optional. If not specified, the Lanman agent discovers the subnet mask from the current configuration. **Note:** MACAddress is required if netbios is disabled for the IP address. Type and Dimension: string-vector

Updating manual DNS entries

Perform the following steps to update the DNS for manual DNS entries.

To update the DNS for manual DNS entries

1. For the manually added DNS entry, add the user in whose context the VCS Helper service is running.
2. Assign "Full Control" privilege to the newly added user.

Refer to Microsoft documentation for information about adding users and assigning privileges.

Updating DNS servers

The following table presents possible combinations of values for the DNSOptions attribute and the updates effected by the Lanman resource corresponding to each value set.

Table 3-7 DNSOptions attribute and Lanman agent behavior

UpdateAll	Ignore Default	Purge Duplicate	Effect
-	-	-	Updates any one default DNS server.
-	-	✓	Updates any one default DNS server and removes duplicate entries, if any.
-	✓	-	Updates any one additional DNS server.
-	✓	✓	Updates any one additional DNS server and removes duplicate entries, if any.
✓	-	-	Updates all the default and additional DNS servers.
✓	-	✓	Updates all the default and additional DNS servers and removes duplicate entries, if any.
✓	✓	-	Updates all the additional DNS servers.
✓	✓	✓	Updates all additional DNS servers and removes duplicate entries, if any.

Sample configurations

In the following example, the Lanman agent is configured to bind a single IP address with the virtual computer name VNAMEX1. The Lanman agent will also add the computer object to the specified Active Directory container and update the additional DNS servers.

```
IP VIP1 (
    Address = "172.29.9.159"
    SubNetMask = "255.255.254.0"
    MACAddress @SYSTEM1 = "00-B0-D0-AB-4E-22"
    MACAddress @SYSTEM2 = "00-B0-D0-D1-69-7D"
)

Lanman VNAMEX1 (
    VirtualName = VNAMEX1
    IPResName = VIP1
    ADContainer = "OU=myUnit,DC=myCompanny,DC=com"
    DnsOptions = { UpdateAll, PurgeDuplicate }
    AdditionalDNSServers = {"10.212.100.20", "10.212.100.21"}
```

```
)

VNAMEX1 requires VIP1
```

In the next example, the Lanman agent is configured to bind multiple IP addresses with the virtual computer name VNAMEX1.

```
IP IP_RES1 (
    Address = "11.212.96.182"
    SubNetMask = "255.255.254.0"
    MACAddress @SYSTEM3 = "00-D0-B7-4A-03-D8"
    MACAddress @SYSTEM4 = "00-D0-B7-4A-04-19"
)

IP IP_RES2 (
    Address = "11.212.99.214"
    SubNetMask = "255.255.254.0"
    MACAddress @SYSTEM3 = "00-D0-B7-4A-03-71"
    MACAddress @SYSTEM4 = "00-D0-B7-4A-03-7D"
)

Lanman LANMAN_VNAMEX1 (
    VirtualName = VNAMEX1
    MultiNet = 1
    MultiNetInfo = {
         "IP=11.212.96.182 MASK=255.255.254.0
        MACAddress=00-D0-B7-4A-03-D8 WINS=10.212.100.120",
         "IP=11.212.99.214 MASK=255.255.254.0
        MACAddress=00-D0-B7-4A-03-D8 WINS=10.212.100.120" }
)

NIC NIC-EXCH_GRP1 (
    MACAddress @SYSTEM3 = "00-D0-B7-4A-03-D8"
    MACAddress @SYSTEM4 = "00-D0-B7-4A-04-19"
    )

NIC NIC-EXCH_GRP2 (
    MACAddress @SYSTEM3 = "00-D0-B7-4A-03-71"
    MACAddress @SYSTEM4 = "00-D0-B7-4A-03-7D"
    )

IP_RES1 requires NIC-EXCH_GRP1
IP_RES2 requires NIC-EXCH_GRP2
```

```
LANMAN_VNAMEX1 requires IP_RES2
LANMAN_VNAMEX1 requires IP_RES1
```

AWSIP agent

The AWSIP agent manages the networking resources in an Amazon Web Services (AWS) cloud environment. The following are the networking resources:

- Private IP - Private IP is a private numerical address that networked devices use to communicate with one another.
- Elastic IP - An Elastic IP address is a static IPv4 public address designed for dynamic cloud computing. An Elastic IP address is associated with your AWS account.
- Overlay IP - AWS allows you to redirect IP address traffic to an Elastic Compute Cloud (EC2) instance in a Virtual Private Network (VPC) no matter which subnet or availability zone (AZ) it is in. Overlay IP provides IP failover functionality for nodes spread across subnets or availability zones. Overlay IP must be outside of the VPC Classless Inter-Domain Routing (CIDR) block.

The agent uses AWS CLIs to associate IP resources in an AWS cloud environment. The agent does the following:

- Assigns and unassigns private IP address
- Associates and disassociates Elastic IP address and assigns/unassigns private IP
- Manages route table entries of overlay IP for failing over across subnets

Prerequisites

- Install the AWS CLI package.
- Go to **EC2 instance** > **Networking** > **Change Source/Dest**. Check and disable **Change Source/Dest. Check** for overlay IP.
- Create an IAM role with requisite permissions and attach the roles to the EC2 cluster instances.

  Go to **IAM** > **Create Policy** > **Role** and create the role.

  The following is a sample of the `policy.json` file:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
```

```
                    "ec2:AssignPrivateIpAddresses",
                    "ec2:DescribeAddresses",
                    "ec2:DisassociateAddress",
                    "ec2:AssociateAddress",
                    "ec2:UnassignPrivateIpAddresses",
                    "ec2:AssignPrivateIpAddresses",
                    "ec2:AssignIpv6Addresses",
                    "ec2:UnassignIpv6Addresses",
                    "ec2:DescribeInstances",
                    "ec2:DescribeNetworkInterfaces",
                    "ec2:DescribeRouteTables",
                    "ec2:CreateRoute",
                    "ec2:DeleteRoute",
                    "ec2:ReplaceRoute"
                ],
                "Effect": "Allow",
                "Resource": "*"
            }
        ]
    }
```

- Ensure that all outgoing traffic goes through network address translation (NAT) gateways. The agent uses AWS CLIs and requires access to the Amazon EC2 API endpoints.
- Ensure that the subnets in which the EC2 exists are associated with the route table.

Dependency

AWSIP resources depend on IP resources.

Agent functions

Online

- Private IP: Assigns the private IP to a NIC Elastic Network Interface (ENI).
- Elastic IP: Works only if a private IP is defined in the configuration. Associates the private IP with an elastic IP. Agent automatically detects the ENI based on the Device attribute.
- Overlay IP: Creates a route in the route table given in the RouteTableIds attribute with destination as the OverlayIP and target as the ENI on which the entry point is running.

Offline	■ Private IP: Unassigns from the ENI. ■ Elastic IP: Disassociates the elastic IP. ■ Overlay IP: Deletes the route from the route table.
Monitor	■ Private IP: Monitors the assignment of the private IP to the ENI. ■ Elastic IP: Monitors the association between private IP and elastic IP. ■ Overlay IP: Monitors the routing information and checks whether the route table entry exists for the overlay IP.
Clean	■ Private IP: Unassigns the private IP. ■ Elastic IP: Disassociates the elastic IP and unassigns the private IP address. ■ Overlay IP: Deletes the route from the route table.

State definitions

ONLINE	■ Private IP: Private IP is assigned to the ENI. ■ Elastic IP: Private IP is assigned and elastic IP is associated with the private IP. ■ Overlay IP: Route table entry exists for the overlay IP.
OFFLINE	■ Private IP: Private IP is not assigned to the ENI. ■ Elastic IP: Private IP is not assigned or elastic IP is not associated with the private IP. ■ Overlay IP: Route table entry does not exist for the overlay IP.
UNKNOWN	If private IP, elastic IP, or overlay IP is in UNKNOWN state, one of the following could be true: ■ AWS CLI is not installed. ■ AWSBinDir attribute is not configured correctly. That is, the AWSBinDir attribute is not set to the location where AWS CLI is installed. For example, `C:\Program Files\Amazon\AWSCLI`. ■ Roles are not configured correctly or attached to the instance.
FAULTED	Indicates that the IP resources could not be brought online or abruptly stopped outside of VCS control.

Resource type definition

```
type AWSIP (
    static i18nstr ArgList[] = { PrivateIP, OverlayIP, ElasticIP,
```

```
      Device, RouteTableIds, AWSBinDir }
    str PrivateIP
    str OverlayIP
    str ElasticIP
    str Device
    keylist RouteTableIds
    str AWSBinDir
)
```

Attribute definitions

Table 3-8 Required attributes

Attribute	Description
PrivateIP	Secondary private IP address of the EC2 instance. Type and dimension: string-scalar
OverlayIP	Overlay IP provides IP failover functionality for nodes spread across subnets or availability zones. Overlay IP must be outside of the VPC CIDR block in which the nodes are present. Type and dimension: string-scalar

Table 3-9 Optional attributes

Attribute	Description
ElasticIP	An Elastic IP address is a static IPv4 address designed for dynamic cloud computing. An Elastic IP address is associated with your AWS account. You can map the Elastic IP address to a secondary private IP address. **Note:** IPv6 addresses are not supported with this attribute. Type and dimension: string-scalar

Table 3-9 Optional attributes *(continued)*

Attribute	Description
Device	The physical address of the interface or the interface name of the NIC to which the virtual IP address to be monitored is assigned. Note that this attribute is always local, that is, it is different for each system. If you change the interface name, you must update the attribute value to specify the new name. When you edit the MACAddress attribute of the IP or the NIC agent to specify the interface name, you must specify the name of only one interface. For more details on modifying the default value of the attribute, refer to the Cluster Server Administrator's Guide. **Note:** Use the `ipconfig -all` command to retrieve the physical address of a NIC. Type and Dimension: string-scalar
RouteTableIds	Describes all the route tables. It can be one or more route table IDs. Type and dimension: string-list
AWSBinDir	Location of AWS EC2 commands and binaries. Type and dimension: string-scalar

Sample configuration

Private IP

```
AWSIP privateip (
 PrivateIP = "10.239.3.97"
 Device@WIN-38PNEVJSR2K = "12-7F-CE-5B-E2-6E"
 Device@WIN-75PQEVSRV2K = "13-7F-CE-6B-E2-6E"
 )
```

Elastic IP

```
AWSIP elasticip (
 PrivateIP = "10.239.3.95"
 ElasticIP = "34.193.196.156"
 Device @WIN-38PNEVJSR2K = "12-7F-CE-5B-E2-6E"
 Device @WIN-42PNEVXYR5K = "18-6F-BE-9B-E2-6E"
 )
```

Overlay IP

```
AWSIP overlayip (
 OverlayIP = "172.16.8.55/32"
 Device @WIN-38PNEVJSR2K = "12-7F-CE-5B-E2-6E"
 Device @WIN-45ABCVJSR2K = "14-7E-DE-6B-E2-8E"
 RouteTableIds = { rtb-c5272ca3, rtb-fb97ac9d }
 )
```

AzureIP agent

The AzureIP agent manages the following networking resources in an Azure environment:

- Private IP—A private IP is a private numerical address that networked devices use to communicate with one another. It is used for communication between an Azure virtual network (VNet) and an on-premises network. To extend an on-premises network to Azure, you use a VPN gateway or an ExpressRoute circuit.
- Public IP— A public IP is a numerical address that is used for communication over the internet, including Azure public-facing services.
- Overlay IP—An overlay IP provides IP failover functionality for nodes spread across subnets. It allows you to redirect IP address traffic to another cluster node belonging to different subnet within the same VNet. An overlay IP must be defined outside the VNet Classless Inter-Domain Routing (CIDR) block.

  Note: An overlay IP cannot be used to provide failover across Regions if VNet-to-VNet tunneling is used.

The AzureIP agent performs the following activities:

- Gets the NIC details, creates the IP configuration, and associates and disassociates the private IP address
- Associates and disassociates the Public IP address with the Private IP address
- Manages route table entries of Overlay IP for failover across subnets

The AzureIP agent uses Azure Python APIs to associate IP resources with an Azure VM.

An InfoScale deployment in Azure does not support IPv6 because of the following limitations:

- Existing virtual machines (VMs) that are configured with IPv4 addresses cannot use IPv6 addresses; you must deploy new VMs and configure them with IPv6 addresses.
- Public IPv6 addresses cannot be assigned to a VM.
- VMs with IPv6 addresses cannot be members of an Azure cloud service. However, they can communicate with each other over their respective IPv4 addresses.

Prerequisites

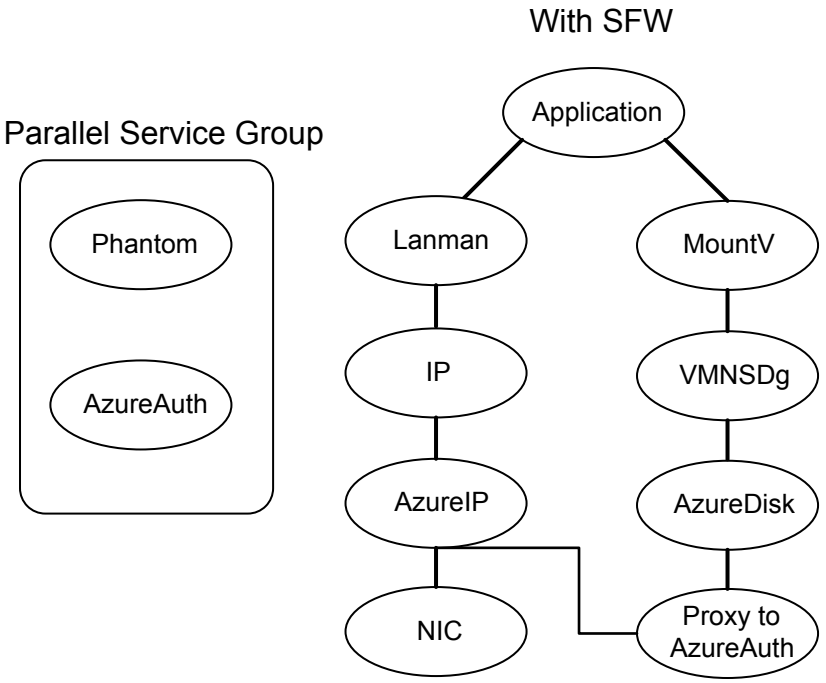
- Configure the AzureAuth agent.
  See "AzureAuth agent" on page 151.
- To configure a public IP, create a static public IP resource in the Azure portal.
- To configure an overlay IP, create a route table, and associate the subnets where the overlay IP fails over with the route table.

Note: The subnets can be associated with a single route table.

Dependencies

The AzureIP resources depend on the AzureAuth resources.

Figure 3-1 Sample service group dependency



Agent functions

Online

- Private IP: Creates the IP configuration with the private IP and associates it with the Network Interface.
- Public IP: Creates the IP configuration with the private and public IP. Associates the IP configuration with the Network Interface.
- Overlay IP: Creates a route in the Azure route table given in the RouteTableResourceIds attribute with Address Prefix as the OverlayIP and next hop as the private IP of the primary NIC.

- Offline
- Clean

- Private IP: Deletes the IP configuration to disassociate Private IP from the Network Interface.
- Public IP: Deletes the IP configuration to disassociate the Public IP along with the Private IP from the Network Interface.
- Overlay IP: Deletes the route from the route table.

Monitor

- Private IP: Monitors the associations between IP configuration and Network Interface.
- Public IP: Monitors the association between IP configuration and Network Interface.
- Overlay IP: Monitors the routing information and checks whether the route table entry exists for the overlay IP.

State definitions

ONLINE

- Private IP: Indicates that the Private IP is assigned to the NIC.
- Public IP: Indicates that the Private IP is assigned and Public IP is associated with the private IP.
- Overlay IP: Indicates that Route table entry exists for the overlay IP.

OFFLINE

- Private IP: Indicates that the Private IP is not assigned to the NIC.
- Public IP: Indicates that the Private IP is not assigned or Public IP is not associated with the private IP.
- Overlay IP: Indicates that Route table entry does not exist for the overlay IP.

UNKNOWN

One of the following could be true:

- Azure Python SDK is not installed.
- Required privileges are not present to perform operations on Azure NIC and Virtual Machine.
- Private IP, Public IP, or Overlay IP is specified in wrong format.
- Private IP, Public IP, or Overlay IP is already in use.
- MACAddress attribute value is invalid.
- In case of Overlay IP, route table id(s) are invalid.
- IP Configuration has invalid details on NIC.
- Both or none of the Overlay IP and Private IP are set. Only one should be set.

FAULTED

Indicates that the IP resources could not be brought online or abruptly stopped outside of VCS control.

Attributes

Table 3-10 Required attributes

Attribute	Description
PrivateIP	Secondary private IP address of the Azure VM. This value is mandatory if OverlayIP is not provided. Type and dimension: string-scalar

Table 3-10 Required attributes *(continued)*

Attribute	Description
MACAddress	The physical address of the network interface to which the virtual IP address to be monitored is assigned. Note that this attribute is always local, that is, it is different for each system. Use the `ipconfig -all` command to retrieve the physical address of a NIC. Example: 00-60-08-08-BA-C4 Type and Dimension: string-scalar
OverlayIP	Overlay IP provides IP failover functionality for nodes spread across subnets. Overlay IP must be outside of the VNet CIDR block in which the nodes are present. This value is mandatory if PrivateIP is not provided. Type and dimension: string-scalar
RouteTableResourceIds	Describes all route tables. You can add one or more route table IDs separated by a space. Type and dimension: string-vector
AzureAuthResName	Name of the authentication agent resource that handles Azure related authentication. Type and dimension: string-scalar

Table 3-11 Optional attributes

Attribute	Description
PublicIP	Static public IP created from the Azure Portal. This IP is used in IP configuration to map the Public IP address to a secondary private IP address. Type and dimension: string-scalar
AzureVMName	Name of the VM in Azure on which agent is running. Type and dimension: string-scalar
VMResourceGroup	Azure resource group where the Azure VM resides. Type and dimension: string-scalar

Resource type definition

```
type AzureIP (
    static i18nstr ArgList[] = { tempVMName, PrivateIP, MACAddress,
      PublicIP, tempPublicIPResourceId, AzureVMName, VMResourceGroup,
      OverlayIP, RouteTableResourceIds, "AzureAuthResName:SubscriptionId",
      "AzureAuthResName:ClientId", "AzureAuthResName:SecretKey",
      "AzureAuthResName:TenantId", tempVMResourceGroupName }
    str PrivateIP
    str MACAddress
    str PublicIP
    temp str tempPublicIPResourceId
    str AzureVMName
    str VMResourceGroup
    str OverlayIP
    str RouteTableResourceIds[]
    str AzureAuthResName
    temp str tempVMName
    temp str tempVMResourceGroupName
)
```

Sample configurations

Sample configuration with Private IP:

```
AzureIP azure-ip-res (
    PrivateIP = "10.1.5.42"
    MACAddress @cloudvm1 = 00-0D-3A-91-09-87
    MACAddress @cloudvm2 = 12-0D-3B-87-19-67
    AzureAuthResName = Auth_Res
)
```

Sample configuration with Public IP:

```
AzureIP azure-ip-res (
    PrivateIP = "10.1.5.52"
    MACAddress @cloudvm1 = 00-0D-3A-91-09-87
    MACAddress @cloudvm2 = 12-0D-3B-87-19-67
    PublicIP = "52.173.243.126"
    AzureAuthResName = Auth_Res
)
```

Sample configuration with Overlay IP:

```
AzureIP overlay-ip-res (
    MACAddress @cloudvm1 = 00-0D-3A-91-09-87
```

```
    MACAddress @cloudvm2 = 12-0D-3B-87-19-67
    OverlayIP = "192.168.3.88"
    RouteTableResourceIds = {
      "/subscriptions/6940a326-abc6-40dd-b616-d3f9bbdf1d63/
      resourceGroups/azureRG/providers/Microsoft.Network/
      routeTables/azureroute1",
      "/subscriptions/6940a326-abc6-40dd-b616-d3f9bbdf1d63/
      resourceGroups/azureRG/providers/Microsoft.Network/
      routeTables/azureroute2"}
    AzureAuthResName = Auth_Res
)
```

Debug log levels

The AzureIP agent uses the DBG_1 and the DBG_2 debug log levels.

AzureDNSZone agent

Azure DNS is a hosting service for DNS domains, providing name resolution using Microsoft Azure infrastructure. Azure DNS lets you host your DNS domains on the cloud using Microsoft Azure infrastructure. The AzureDNSZone agent monitors and updates the host name to resource record mapping. The agent does the mapping for the Azure DNS domain when failing over nodes across subnets or regions. To start hosting your domain in Azure DNS, you need to create a DNS zone for that domain name. When you create a DNS zone, Azure DNS automatically creates a name server (NS) record and a start of authority (SOA) record for the zone.

AzureDNSZone agent provides DNS-based traffic routing and failover. Use the AzureDNSZone agent if the resource records need to be dynamically added and deleted from the domain during failover. The agent updates the new resource record mappings while failing over and allows the clients to connect to the failed over instance of the application.

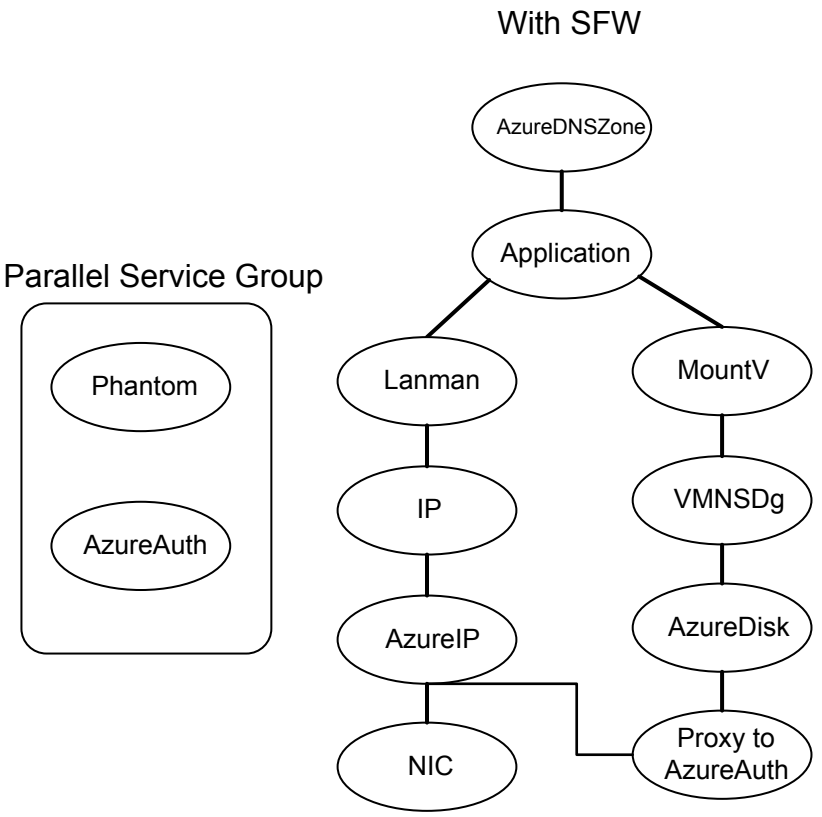
Prerequisites

- Configure AzureAuth agent. See “AzureAuth agent” on page 151.
- Own a domain name
- Create DNS zone with root domain name
- Delegate the domain name to Azure DNS. See “Delegating a domain to Azure DNS” on page 98.

Dependencies

The AzureDNSZone resources depend on the AzureAuth resource.

Figure 3-2 Sample service group dependency



Agent functions

Online Creates or updates resource record sets in Azure DNS Zone.

Offline	Removes the resource record set from the hosted Azure DNS Zone if OffDelRR is set to 1. If OffDelRR is not set to 1, records are not deleted. The default value of OffDelRR is 0.
Monitor	Monitors if the resource record sets are present, and if all mappings are present in the hosted Azure DNS Zone. **Note:** If resource record sets are added or deleted manually, the agent does not monitor the change in state of these record sets.
Clean	Removes the resource record set from the hosted Azure DNS Zone if OffDelRR is set to 1. If OffDelRR is not set to 1, records are not deleted.

State definitions

ONLINE	Indicates that all the resource record sets are present in the Azure DNS Zone.
OFFLINE	Indicates that all or one of the resource record sets is not present in the Azure DNS Zone.
UNKNOWN	Indicates that a problem exists because of one of the following reasons: ■ Azure Python SDKs are not installed ■ Azure DNS Zone is not present ■ Roles are not configured properly ■ AzureDnsZoneResourceId is invalid ■ Resource Record Type (RSType) is not A, AAAA, CNAME, or PTR ■ ResRecord value does not have appropriate RSType value

Attributes

Table 3-12 Required attributes

Attribute	Description
AzureDnsZoneResourceId	Resource ID of the Azure DNS Zone. Type and dimension: string-scalar

Table 3-12 Required attributes *(continued)*

Attribute	Description
ResRecord	ResRecord is an association of DNS resource record values. Each ResRecord attribute consists of two values: DNS record key and 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. Resource records can be of the following types: ■ A ■ AAAA ■ CNAME ■ PTR Type and dimension: string-association
RSType	Record types supported by Azure DNS Zone. Type and dimension: string-scalar
AzureAuthResName	Name of the authentication agent resource that handles Azure related authentication. Type and dimension: string-scalar

Table 3-13 Optional attributes

Attribute	Description
TTL	Specifies the minimum time to live (in seconds) for all resource records. Default value is 3600 seconds. Type and dimension: int-scalar
OffDelRR	Determines if resource records will be deleted as part of the resource offline entry point. Default is false. Type and dimension: boolean-scalar

Resource type definition

```
type AzureDNSZone (
        static i18nstr ArgList[] = { AzureDnsZoneResourceId, RSType, ResRecor
        "AzureAuthResName:SubscriptionId", "AzureAuthResName:ClientId",
        "AzureAuthResName:SecretKey", "AzureAuthResName:TenantId" }
```

```
                    str AzureDnsZoneResourceId
                    str RSType
                    str ResRecords{}
                    int TTL = 3600
                    boolean OffDelRR = 0
                    str AzureAuthResName
                )
```

Samples configurations

A

```
AzureDNSZone a_record (
AzureDnsZoneResourceId = "/subscriptions/6940a326-fgh6-40dd-b616-c1e9bbdf1d63/resourceGroups/
                    azureRG/providers/Microsoft.Network/dnszones/example.com
ResRecords = { "www" = "10.44.50.1,10.45.1.1", "vm1" = "10.44.50.2", "vm2" = "10.44.51.10",
             "@" = "10.45.55.66" }
RSType = "A"
TTL = 300
AzureAuthRes = "azure-auth-res"
)
```

AAAA

```
AzureDNSZone aaaa_record (
    AzureDnsZoneResourceId = "/subscriptions/6940a326-fgh6-40dd-b616-c1e9bbdf1d63/
      resourceGroups/resgrp/roviders/Microsoft.Network/dnszones/sampledomain.com"
    ResRecords = { "www" = " 2001:0db8:85a3:0000:0000:8a2e:0370:7334",
                   "vm1" = " 2607:f0d0:1002:0051:0000:0000:0000:0004",
                   "@" = "2001:0db8:85a3:0000:0000:8a2e:0370:7334" }
    RSType = "AAAA"
    TTL = 300
    AzureAuthRes = "azure-auth-res"
    )
```

CNAME

```
AzureDNSZone cname_record (
    AzureDnsZoneResourceId = "/subscriptions/6940a326-fgh6-40dd-b616-c1e9bbdf1d63/
       resourceGroups/resgrp/providers/Microsoft.Network/dnszones/sampledomain.com"
    ResRecords = { "vm1" = "vm1.alias.com", "vm2" = "vm2.alias.com" }
    RSType = "CNAME"
    TTL = 300
```

```
    AzureAuthRes = "azure-auth-res"
    )
```

PTR

```
AzureDNSZone ptr_record (
    AzureDnsZoneResourceId = "/subscriptions/6940a326-fgh6-40dd-b616-c1e9bbdf1d63/
       resourceGroups/resgrp/providers/Microsoft.Network/dnszones/sampledomain_ptr.com"
    ResRecords = { "11" = "vm1.alias.com", "222" = "vm2.alias.com" }
    RSType = "PTR"
    TTL = 300
    AzureAuthRes = "azure-auth-res"
    )
```

In this sample for PTR, 11 and 222 in ResRecords indicates the last block of the IP address. For example, 11 and 222 in 192.168.10.11 and 192.168.10.222 respectively.

Delegating a domain to Azure DNS

Delegate a domain to Azure DNS

1. Log on to the Azure portal.
2. Create a DNS zone.

   Azure DNS allocates name servers names for your zone.
3. In the **Favorites** pane, click **All resources** and get the name servers names from the DNS zone.
4. Using the DNS management tools of your existing registrar, edit the NS records and replace the NS records with the ones allocated by Azure DNS.

   Note: When delegating a domain to Azure DNS, you must use the name server names provided by Azure DNS. It is recommended to use all four name server names, regardless of the name of your domain. Domain delegation does not require the name server name to use the same top-level domain as your domain.
5. After completing the delegation, verify that the name resolution is working.

   If the delegation is set up correctly, the normal DNS resolution process finds the name servers automatically.
6. If the name resolution is not working, update the start of authority (SOA) record allocated by Azure DNS in your existing registrar.

GoogleIP agent

The GoogleIP agent manages the following networking resources in a Google Cloud Platform (GCP) environment:

- Private IP—A private IP is a private numerical address that networked devices use to communicate with one another. It is used for communication between a GCP virtual private cloud (VPC) network and an on-premises network. A private IP is also referred to as Alias IP in GCP. To extend an on-premises network to a GCP VPC network, you use a VPN gateway.
- Overlay IP—An overlay IP provides IP failover functionality for the InfoScale cluster nodes that are spread across subnets. It lets you redirect IP address traffic to another cluster node belonging to different subnet within the same VPC. An overlay IP must be defined outside the VPC Classless Inter-Domain Routing (CIDR) block.

The GoogleIP agent performs the following activities:

- Fetches the NIC details, associates the private IP with the NIC, and disassociates the private IP from the NIC
- Manages the route table entries of the overlay IP for failover across subnets

The GoogleIP agent uses GCP Python APIs to associate IP resources with a GCP VM instance.

Note: An InfoScale deployment in GCP does not support IPv6 because VPC networks in GCP do not support IPv6 traffic within the network. For details, refer to the GCP documentation on VPC networks.

Prerequisites

- The Google service account associated with the VM instance have the following roles assigned, at a minimum:
  - compute.globalOperations.get
  - compute.instances.get
  - compute.instances.updateNetworkInterface
  - compute.networks.updatePolicy
  - compute.projects.get
  - compute.routes.create
  - compute.routes.delete
  - compute.routes.get

- compute.routes.list
- compute.zoneOperations.get

- To access Google APIs, the following Python modules must be present on the InfoScale cluster nodes:
  - `oauth2client`
  - `google-api-python-client`

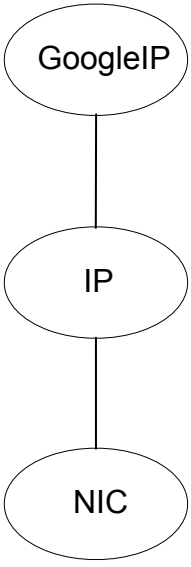
Install Python SDK for GCP on each of the cluster nodes by running the following commands at the location `C:\Program Files\Veritas\VRTSPython`, or the location where the `VRTSPython` folder is installed on the node:

- python -m pip install --upgrade pip
- python -m pip install --upgrade google-api-python-client
- python -m pip install --upgrade oauth2client

Dependencies

A GoogleIP resource depends on the IP and the NIC resources.

Figure 3-3 Sample service group dependency



Agent functions

Online	■ Private IP: Associates the private IP with the network interface. ■ Overlay IP: Creates a route in route table with the required base IP. Sets the overlay IP as the destination and the private IP as the next hop of the primary NIC.
Offline and Clean	■ Private IP: Disassociates the private IP from the network interface. ■ Overlay IP: Deletes the route from the route table.
Monitor	■ Private IP: Monitors the association between the private IP and the network interface. ■ Overlay IP: Monitors the routing information and checks whether the route table entry exists for the overlay IP.

State definitions

ONLINE	■ Private IP: Indicates that the private IP is assigned to the NIC. ■ Overlay IP: Indicates that route table entry exists for the overlay IP.
OFFLINE	■ Private IP: Indicates that the private IP is not assigned to the NIC. ■ Overlay IP: Indicates that route table entry does not exist for the overlay IP.
UNKNOWN	This state indicates that one of the following situations may have occurred: ■ The required GCP Python SDKs are not installed. ■ The privileges that are needed to perform operations on the NIC and the VM instance are not provided. ■ The private IP or the overlay IP is specified in an incorrect format. ■ MACAddress attribute value is invalid. ■ The route that is associated with the overlay IP in the route table is missing. ■ Either overlay IP and private IP are both set, or none of them are set—only one of them should be set at a time.
FAULTED	Indicates that the IP resources cannot be brought online or were abruptly stopped outside of VCS control.

Limitation

When an IP address that is associated with an OverlayIP resource is already online elsewhere in the VPC network, the GoogleIP service group with that OverlayIP resource should not come online. However, the service group does come online, and then the IP resource faults in both the clusters.

Workaround: Do not configure the IP address that is associated with the OverlayIP resource for any other device within the same VPC network.

Attributes

Table 3-14 Required attributes

Attribute	Description
PrivateIP	Secondary private IP address, also known as Alias IP, of the Google VM. You must specify a value for this attribute if no value is specified for OverlayIP. Type and dimension: string-scalar
MACAddress	The physical address of the network interface to which the virtual IP address to be monitored is assigned. Note that this attribute is always local, that is, it is different for each system. Use the `ipconfig -all` command to retrieve the physical address of a NIC. Example: 00-60-08-08-BA-C4 Type and Dimension: string-scalar
OverlayIP	Overlay IP provides IP failover functionality for the InfoScale cluster nodes that are spread across subnets. Overlay IP must be outside of the VPC CIDR block in which the nodes are present. You must specify a value for this attribute if no value is specified for PrivateIP. Type and dimension: string-scalar

Table 3-15 Optional attributes

Attribute	Description
ProjectID	The customized name that you choose when you create a GCP project, or when you activate an API that requires you to create a project ID. Type and dimension: string-scalar
VMName	Name of the GCP VM instance on which the agent is running. Type and dimension: string-scalar

Table 3-15 Optional attributes *(continued)*

Attribute	Description
Zone	The GCP zone to which the VM instance belongs. Type and dimension: string-scalar

Resource type definition

```
type GoogleIP (
    static keylist RegList = { MACAddress }
    static i18nstr ArgList[] = { PrivateIP, MACAddress, OverlayIP,
      ProjectID, Zone, VMName, tempProjectName, tempVMName,
      tempZoneName, tempRouteName, tempDeviceName }
    temp str tempProjectName
    temp str tempRouteName
    temp str tempZoneName
    temp str tempVMName
    temp str tempDeviceName
    str PrivateIP
    str MACAddress
    str OverlayIP
    str ProjectID
    str Zone
    str VMName
)
```

Sample configurations

Sample configuration with Private IP:

```
GoogleIP csg_GoogleIP (
    PrivateIP = "10.208.8.21"
    MACAddress @cloudvm1 = "42:01:0A:D0:08:06"
    MACAddress @cloudvm2 = "42:01:0A:D0:08:05"
)

IP csg_ip (
    Address = "10.208.8.21"
    SubNetMask = "255.255.252.0"
    MACAddress @cloudvm1 = "42:01:0A:D0:08:06"
    MACAddress @cloudvm2 = "42:01:0A:D0:08:05"
)

NIC csg_nic (
```

```
    MACAddress @cloudvm1 = "42:01:0A:D0:08:06"
    MACAddress @cloudvm2 = "42:01:0A:D0:08:05"
)

csgip requires csgnic
csg_GoogleIP requires csgip
```

Sample configuration with Overlay IP:

```
GoogleIP csg_GoogleIP (
    OverlayIP = "192.168.9.3"
    MACAddress @cloudvm1 = "42:01:0A:D0:08:06"
    MACAddress @cloudvm2 = "42:01:0A:D0:08:05"
)

IP csg_ip (
    Address = "192.168.9.3"
    SubNetMask = "255.255.252.0"
    MACAddress @cloudvm1 = "42:01:0A:D0:08:06"
    MACAddress @cloudvm2 = "42:01:0A:D0:08:05"
)

NIC csg_nic (
    MACAddress @cloudvm1 = "42:01:0A:D0:08:06"
    MACAddress @cloudvm2 = "42:01:0A:D0:08:05"
)

csgip requires csgnic
csg_GoogleIP requires csgip
```

Debug log levels

The GoogleIP agent uses the DBG_1 debug log level.

Modules tested

Veritas has tested and approved the GoogleIP agent with the following modules:

GCP Python module	Version
cachetools	2.1.0
google-api-python-client	1.7.4
google-auth	1.6.1
google-auth-httplib2	0.0.3

GCP Python module	Version
httplib2	0.11.3
oauth2client	4.1.3
pyasn1	0.4.4
pyasn1-modules	0.2.2
rsa	4.0
six	1.11.0
uritemplate	3.0.0

Chapter 4

FileShare agents

This chapter includes the following topics:

- About FileShare agents
- FileShare agent
- CompositeFileShare agent

About FileShare agents

FileShare agents make file shares highly available. The FileShare agents include the FileShare and the CompositeFileShare agent. The FileShare agent ensures high availability for a shared folder and its subfolders. The CompositeFileShare agent ensures high availability for multiple subfolders.

FileShare agent

The FileShare agent enables systems to share a folder and its subfolders, making the shared folder highly available. The folder to be shared is stored on a shared disk.

Using the FileShare agent you can also do the following:

- Create hidden shares for a specific share or subfolders.
- Dynamically share subfolders created after the resource was brought online.
- Configure and control existing shares.

The FileShare agent enables sharing folders shared outside VCS. However, you cannot add special shares (shares created by the operating system for administrative and system use) to the VCS configuration. For example, you cannot add the shares ADMIN$, print$, IPC$, and *DriveLetter*$ to the VCS configuration.

You can configure a file share service group using the File Share Configuration Wizard.

See the *Cluster Server Administrator's Guide* for instructions.

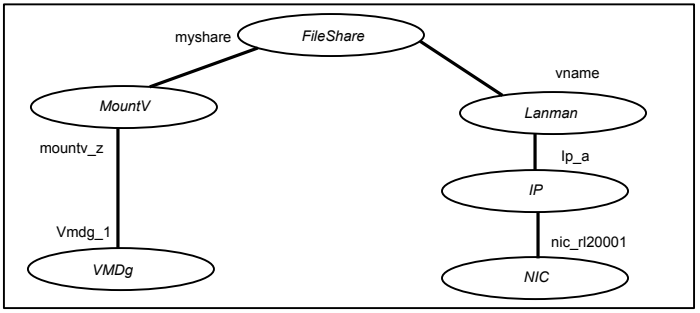
To share multiple subfolders, use the CompositeFileShare agent.

See "CompositeFileShare agent" on page 116.

Dependencies

The FileShare resource depends on the MountV resource. A FileShare service group also requires a Lanman resource that is dependent on the IP resource, as illustrated in the following dependency graph.

Figure 4-1 File Share service group dependency graph



Note: In case of a NetApp storage environment, use the NetApp Filer and NetApp SnapDrive agents instead of the VMDg and MountV agents respectively. If you are using Logical Disk Management (LDM) for configuring your storage, use the Mount and the DiskRes agents.

Agent functions

Online	Shares the specified folders with designated permissions.
Offline	Removes the shares for the specified folders.
Monitor	Verifies at least one specified folder is shared with the designated permissions.

State definitions

ONLINE	Indicates at least one specified folder is shared.
OFFLINE	Indicates no specified folder is shared.
UNKNOWN	Indicates the agent could not determine the status of the resource.

Note: Sharing a folder with a large number of subfolders and enabling the ShareSubdirectories attribute may cause increased failover time and high CPU and memory utilization.

About ForceControl function

ForceControl is a FileShare agent function that allows you to reset the file share properties as per what is defined in the VCS configuration. The agent overwrites all modifications made to the file share properties, either externally or from within VCS, and sets the properties as per what you had defined while configuring the file shares in VCS. This function overrides the FileShare agent's AutoControl attribute settings.

You can run the ForceControl function from the command line. To run the ForceControl function on a fileshare resource, type the following command:

```
hares -action <fileshare_resource_name> ForceControl
-sys <system_name>
```

Here, *<fileshare_resource_name>* is the name of the file share resource for which you want to reset the properties and *<system_name>* is the name of the cluster node on which that file share resource is online.

For example, to run the ForceControl function on a FileShare resource *A_fileshare* that is currently online on a cluster node *Node1*, type the following:

```
hares -action A_fileshare ForceControl -sys Node1
```

The changes take effect on the next agent monitor function.

Resource type definition

```
type FileShare
static keylist SupportedActions = { ForceControl }
static i18nstr ArgList[] = { PathName, ShareName,
"LanmanResName:VirtualName", MaxUsers, "MountResName:MountPath",
HiddenShare, ShareSubdirectories, HideChildShares, AutoShare,
```

```
AutoControl, ClientCacheType, UserPermissions,
AccessBasedEnumeration, ShareComment }
    i18nstr PathName
    i18nstr ShareName
    str LanmanResName
    str MaxUsers
    str MountResName
    i18nstr UserPermissions{} = { Everyone=READ_ACCESS }
    boolean HiddenShare = 0
    boolean ShareSubdirectories = 0
    boolean HideChildShares = 0
    boolean AutoShare = 1
    boolean AutoControl = 1
    str ClientCacheType = MANUAL
    boolean IgnorePermissions = 0
    boolean AccessBasedEnumeration = 0
    str ShareComment
)
```

Attribute definitions

Table 4-1 FileShare agent required attributes

Required Attributes	Description
LanmanResName	The name of the Lanman resource on which the FileShare resource depends. If this attribute is not specified, the agent ignores this attribute and uses localhost. Type and Dimension: string-scalar
MountResName	The name of the MountV resource on which the FileShare resource depends. For NetApp storage, this is the NetAppSnapDrive resource name. In case of Logical Disk Management, this is the Mount resource name. Type and Dimension: string-scalar

Table 4-1 FileShare agent required attributes *(continued)*

Required Attributes	Description
PathName	The path of the folder to be shared. To share a drive, specify the PathName as \. For example, to share drive X:, the PathName is \. To share a folder on a mounted drive, specify the PathName as *directoryname*. **Note:** This attribute can take localized values. Type and Dimension: string-scalar
ShareName	The name by which the share is known to clients. **Note:** This attribute can take localized values. Type and Dimension: string-scalar

Table 4-2 FileShare agent optional attributes

Optional Attributes	Description
AutoControl	Defines the agent behavior when share properties are modified (either within or from outside VCS) when the FileShare resource is online. The value 1 indicates that the agent synchronizes the changes made to the share properties with those that were defined while configuring the file share service group in the cluster. The value 0 indicates that the agent does not synchronize the share properties as per what is defined in the cluster configuration. If this attribute is set to 0 and the share properties are modified (either within or from outside VCS), the FileShare resource goes into the `UNKNOWN` state. The changes made to the share properties remain in effect until the resource is in the `UNKNOWN` state. To restore the state, you have to take the FileShare resource offline and bring it online again. When the resource is brought online, the agent restores the share properties as per what is defined in the cluster configuration. Default is 1. To make an existing share highly available, the share name and the share permissions in the configuration file must be the same as those for the file share. Type and Dimension: boolean-scalar
AutoShare	Defines agent behavior when a folder with shared subfolders is added to a VCS file share. The value 1 indicates the agent automatically shares the newly added subfolder in its next monitor cycle. The value 0 indicates the agent does not. Default is 1. This attribute is considered only if the attribute ShareSubdirectories is set to 1. Type and Dimension: boolean-scalar

Table 4-2 FileShare agent optional attributes *(continued)*

Optional Attributes	Description
ClientCacheType	A string that specifies whether the files and programs in the shared folder are cached locally on the client computer when accessed by users. It also specifies how the files are cached. The cached files are then available offline even if users are not connected to the share. **Note:** The agent does not cache the files or programs itself. It sets the value so that the server and client interfaces do the needful. This attribute can have the following values: ■ MANUAL: Indicates that only the files and programs specified by the users are cached. ■ NONE: Indicates that files and programs from the share are not cached. ■ DOCS: Indicates that all the files and programs that the users open from the share are automatically cached. Files and programs that are not opened are not available offline. ■ PROGRAMS: Indicates that all files and programs that the users open from the share are automatically cached and are optimized for performance. The next time the user accesses the executable files, they are launched from the local cache. Files and programs that are not accessed are not available offline. Default is MANUAL. Type and Dimension: string-scalar
HiddenShare	Defines whether the agent hides the file share. The value 1 indicates the agent hides the file share. The value 0 indicates it does not. Default is 0. **Note:** To create a hidden share, set the HiddenShare attribute to 1. Do not append the share name with a $ (dollar) sign. Type and Dimension: boolean-scalar

Table 4-2 FileShare agent optional attributes *(continued)*

Optional Attributes	Description
HideChildShares	Defines whether the agent hides the subfolder shares. The value 1 indicates the agent hides the subfolder shares. The value 0 indicates it does not. Default is 0. This attribute is considered only if the attribute ShareSubdirectories is set to 1. Type and Dimension: boolean-scalar
IgnorePermissions	This attribute has been deprecated. Please use AutoControl attribute instead. Type and Dimension: boolean-scalar
MaxUsers	The maximum number of users that can access the file share. Default is null, which indicates access is granted to maximum users allowed on Windows. If this attribute is set to zero or greater than the maximum users allowed on Windows, access is granted to the maximum users allowed on Windows. Type and Dimension: string-scalar
ShareSubdirectories	Defines whether the agent shares the subfolders of the file shares defined in the attribute PathName and ShareName. Subfolders are shared with their own names, that is, the share name of a subfolder is the same as the subfolder name. If a share with the same name exists, the subfolder will not be shared. However, this does not affect the state of the resource. The value 1 indicates the agent shares the subfolders. The value 0 indicates it does not. Default is 0. **Note:** Sharing a folder with a large number of subfolders and enabling the ShareSubdirectories attribute may cause increased failover time and high CPU and memory utilization. Type and Dimension: boolean-scalar

Table 4-2 FileShare agent optional attributes *(continued)*

Optional Attributes	Description
UserPermissions	The permissions with which the folders are shared for users. The following permissions are associated with the FileShare resource: ■ FULL_CONTROL: Permission to read, write, create, execute, and delete the resource, and to modify its attributes and permissions. ■ READ_ACCESS: Permission to read, and execute the resource. ■ CHANGE_ACCESS: Permission to read, write, execute, and delete the resource. ■ NO_ACCESS: No access to the resource. The UserPermissions are specified in the format *'Domain_Name\Username'=Permission*. For example, to give full control to user John who belongs to the domain vcs_domain, the syntax is *'VCS_DOMAIN\John'=FULL_CONTROL*. The domain name and the user name must be enclosed in quotation marks. Default is {'Everyone' = READ_ACCESS}. A maximum of 50 users can be configured for each file share. To configure more than 50 users for a file share, configure user groups. Note that the agent monitors only the users and the permissions that are defined in the VCS configuration. **Note:** This attribute can take localized values. Type and Dimension: string-association
AccessBasedEnumeration	Defines whether the agent enables the Windows Access-based Enumeration option for the specified file share. The value 1 indicates that the agent enables it and the value 0 indicates that the agent does not. Default is 0. Type and Dimension: boolean-scalar

Table 4-2 FileShare agent optional attributes *(continued)*

Optional Attributes	Description
ShareComment	A string that describes the share. Use this attribute to add comments to the file share resource. If this attribute is blank, the agent uses the default VCS comment. Type and Dimension: string-scalar

Sample configuration

Sample file share configuration using the SFW agents (MountV/VMDg):

```
FileShare FileShare_SG-FileShare (
    PathName = "\\ToShare"
    ShareName = ToShare
    LanmanResName = FileShare_SG-Lanman
    MaxUsers = 100
    MountResName = FileShare_SG-MountV
    UserPermissions = { "VCSNET\\Administrator" = READ_ACCESS }
    ShareSubdirectories = 1
    HideChildShares = 1
    AccessBasedEnumeration = 1
    )

IP FileShare_SG-IP (
    Address = "10.212.100.184"
    SubNetMask = "255.255.254.0"
    MACAddress @SYSTEM1 = "00-02-B3-AD-CE-E4"
    MACAddress @SYSTEM2 = "00-02-B3-AD-CB-D0"
    )

Lanman FileShare_SG-Lanman (
    VirtualName = VNAME
    IPResName = FileShare_SG-IP
    )

MountV FileShare_SG-MountV (
    MountPath = "J:"
    VolumeName = PS-SPOOL
    VMDGResName = FileShare_SG-VMDg
    )
```

```
NIC FileShare_SG-NIC (
    MACAddress @SYSTEM1 = "00-02-B3-AD-CE-E4"
    MACAddress @SYSTEM2 = "00-02-B3-AD-CB-D0"
    )

VMDg FileShare_SG-VMDg (
    DiskGroupName = FILESHARE_DG
    DGGuid = e40b8ab5-88ac-42e3-ae61-cbea3a164af2
    )

FileShare_SG-Lanman requires FileShare_SG-IP
FileShare_SG-MountV requires FileShare_SG-VMDg
FileShare_SG-FileShare requires FileShare_SG-Lanman
FileShare_SG-FileShare requires FileShare_SG-MountV
FileShare_SG-IP requires FileShare_SG-NIC
```

CompositeFileShare agent

The CompositeFileShare agent enables systems to share multiple subfolders for each of the shared folders. The folders to be shared are stored on shared disks. Using the CompositeFileShare agent you can also do the following:

- Create hidden shares for shared folders or subfolders.
- Dynamically share the subfolders created after the resource is brought online.
- Configure and control existing shares.

The CompositeFileShare agent enables sharing folders shared outside VCS. However, you cannot add special shares (those created by the operating system for administrative and system use) to the VCS configuration. For example, you cannot add the shares ADMIN$, print$, IPC$, and *DriveLetter*$.

The agent simplifies the process of configuring multiple file shares. It defines file shares in an array, thereby eliminating the need to specify dependencies for each FileShare resource. Each CompositeFileShare resource can be configured for a maximum of 50 file shares.

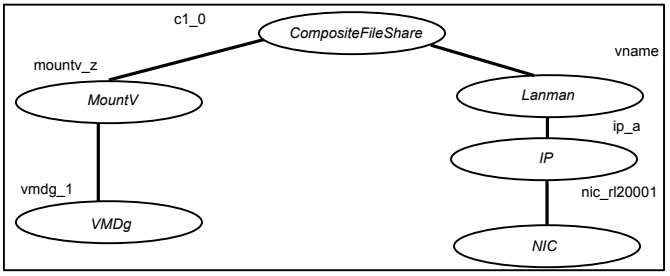
For information on configuring a composite file share service group, refer to the *Cluster Server Administrator's Guide*.

The CompositeFileShare resource depends on the MountV resource. A CompositeFileShare service group also requires a Lanman resource that is dependent on the IP resource.

Dependencies

The CompositeFileShare resource depends on the MountV resource. A CompositeFileShare service group also requires a Lanman resource that is dependent on the IP resource, as illustrated in the following dependency graph.

Figure 4-2 CompositeFileShare service group dependency graph



Note: In case of a NetApp storage environment, use the NetApp Filer and NetApp SnapDrive agents instead of the VMDg and MountV agents respectively. If you are using Logical Disk Management (LDM) for configuring your storage, use the Mount and the DiskRes agents.

Agent functions

Online	Shares the specified folders with designated permissions.
Offline	Removes the shares for the specified folders.
Monitor	Verifies at least one specified folder is shared with the designated permissions.

State definitions

`ONLINE`	Indicates at least one specified folder is shared.
`OFFLINE`	Indicates no specified folder is shared.
`UNKNOWN`	Indicates the agent could not determine the status of the resource.

Note: Sharing a folder with a large number of subfolders and enabling the ShareSubdirectories attribute may cause increased failover time and high CPU and memory utilization.

About ForceControl function

ForceControl is a FileShare agent function that allows you to reset the file share properties as per what is defined in the VCS configuration. The agent overwrites all modifications made to the file share properties, either externally or from within VCS, and sets the properties as per what you had defined while configuring the file shares in VCS. This function overrides the FileShare agent's AutoControl attribute settings.

You can run the ForceControl function from the command line. To run the ForceControl function on a fileshare resource, type the following command:

```
hares -action <fileshare_resource_name> ForceControl
-sys <system_name>
```

Here, *<fileshare_resource_name>* is the name of the file share resource for which you want to reset the properties and *<system_name>* is the name of the cluster node on which that file share resource is online.

For example, to run the ForceControl function on a FileShare resource *A_fileshare* that is currently online on a cluster node *Node1*, type the following:

```
hares -action A_fileshare ForceControl -sys Node1
```

The changes take effect on the next agent monitor function.

Resource type definition

```
type CompositeFileShare
static keylist SupportedActions = { ForceControl }
static i18nstr ArgList[] = { "LanmanResName:VirtualName", MaxUsers,
"MountResName:MountPath", HiddenShare, ShareSubdirectories,
HideChildShares, AutoShare, AutoControl, ClientCacheType,
PathAndShareName, UserPermissions, AccessBasedEnumeration,
ShareComment }
    str LanmanResName
    str MaxUsers
    str MountResName
    i18nstr UserPermissions{} = { Everyone=READ_ACCESS }
    boolean HiddenShare = 0
    boolean ShareSubdirectories = 0
    boolean HideChildShares = 0
    boolean AutoShare = 1
    boolean AutoControl = 1
    i18nstr PathAndShareName{}
    str ClientCacheType = MANUAL
```

```
    boolean IgnorePermissions = 0
    boolean AccessBasedEnumeration = 0
    str ShareComment
)
```

Attribute definitions

Table 4-3 CompositeFileShare agent required attributes

Required Attributes	Description
LanmanResName	The name of the Lanman resource on which the CompositeFileShare resource depends. If this attribute is not specified, the agent ignores this attribute and uses localhost. Type and Dimension: string-scalar
MountResName	The name of the MountV resource on which the CompositeFileShare resource depends. For NetApp storage, this is the NetAppSnapDrive resource name. In case of Logical Disk Management, this is the Mount resource name. Type and Dimension: string-scalar
PathandShareName	A list specifying the respective paths and share names of the folders to be shared. If the path of a shared folder is \Documents, and the share name is UserDocs, the attribute value is defined in the configuration file as {"\\Documents" = "UserDocs"}. While editing the attribute from the Cluster Manager (Java Console) or the command line, specify a single slash for the shared folder. While editing the configuration file main.cf directly, specify two slashes (\\); for example, \\Documents. **Note:** This attribute can take localized values. Type and Dimension: string-association

Table 4-4 CompositeFileShare agent optional attributes

Optional Attributes	Description
AutoControl	Defines the agent behavior when share properties are modified (either within or from outside VCS) when the FileShare resource is online. The value 1 indicates that the agent synchronizes the changes made to the share properties with those that were defined while configuring the file share service group in the cluster. The value 0 indicates that the agent does not synchronize the share properties as per what is defined in the cluster configuration. If this attribute is set to 0 and the share properties are modified (either within or from outside VCS), the FileShare resource goes into the `UNKNOWN` state. The changes made to the share properties remain in effect until the resource is in the `UNKNOWN` state. To restore the state, you have to take the CompositeFileShare resource offline and bring it online again. When the resource is brought online, the agent restores the share properties as per what is defined in the cluster configuration. Default is 1. To make an existing share highly available, the share name and the share permissions in the configuration file must be the same as those for the file share. Type and Dimension: boolean-scalar
AutoShare	Defines agent behavior when a folder is added to a VCS file share with shared subfolders. The value 1 indicates the agent automatically shares the newly added subfolder in its next monitor cycle. The value 0 indicates the agent does not. Default is 1. This attribute is considered only if the attribute ShareSubdirectories is set to 1. Type and Dimension: boolean-scalar

Table 4-4 CompositeFileShare agent optional attributes *(continued)*

Optional Attributes	Description
ClientCacheType	A string that specifies whether the files and programs in the shared folder are cached locally on the client computer when accessed by users. It also specifies how the files are cached. The cached files are then available offline even if users are not connected to the share. **Note:** The agent does not cache the files or programs itself. It sets the value so that the server and client interfaces do the needful. This attribute can have the following values: ■ MANUAL: Indicates that only the files and programs specified by the users are cached. ■ NONE: Indicates that files and programs from the share are not cached. ■ DOCS: Indicates that all the files and programs that the users open from the share are automatically cached. Files and programs that are not opened are not available offline. ■ PROGRAMS: Indicates that all the files and programs that the users open from the share are automatically cached and are optimized for performance. Files and programs that are not opened are not available offline. Default is MANUAL. Type and Dimension: string-scalar
HiddenShare	Defines whether the agent hides the file share. The value 1 indicates the agent hides the file share. The value 0 indicates it does not. Default is 0. **Note:** To create a hidden share, set the HiddenShare attribute to 1. Do not append the share name with a $ (dollar) sign. Type and Dimension: boolean-scalar
HideChildShares	Defines whether the agent hides the subfolder shares. The value 1 indicates the agent hides the subfolder shares. The value 0 indicates it does not. Default is 0. This attribute is considered only if the attribute ShareSubdirectories is set to 1. Type and Dimension: boolean-scalar

Table 4-4 CompositeFileShare agent optional attributes *(continued)*

Optional Attributes	Description
IgnorePermissions	This attribute has been deprecated. Please use AutoControl attribute instead. Type and Dimension: boolean-scalar
MaxUsers	The maximum number of users that can access the file share. Default is null, which indicates access is granted to maximum users allowed on Windows. If this attribute is set to zero or greater than the maximum users allowed on Windows, access is granted to the maximum users allowed on Windows. Type and Dimension: string-scalar
ShareSubdirectories	Defines whether the agent shares the subfolders of the file shares defined in the attribute PathAndShareName. Subfolders are shared with their own names, that is, the share name of a subfolder is the same as the subfolder name. If a share with the same name exists, the subfolder will not be shared. However, this does not affect the state of the resource. The value 1 indicates the agent shares the subfolders. The value 0 indicates it does not. Default is 0. **Note:** Sharing a folder with a large number of subfolders and enabling the ShareSubdirectories attribute may cause increased failover time and high CPU and memory utilization. Type and Dimension: boolean-scalar

Table 4-4 CompositeFileShare agent optional attributes *(continued)*

Optional Attributes	Description
UserPermissions	The permissions with which the folders are shared for users. The user permissions do not apply to individual file shares. User permissions apply to all file shares that form the CompositeFileShare resource. The following permissions are associated with the CompositeFileShare resource: ■ FULL_CONTROL: Permission to read, write, create, execute, and delete the resource, and to modify its attributes and permissions. ■ READ_ACCESS: Permission to read, and execute the resource. ■ CHANGE_ACCESS: Permission to read, write, execute, and delete the resource. ■ NO_ACCESS: No access to the resource. The UserPermissions are specified in the format *'Domain_Name\Username'=Permission*. For example, to give full control to user John who belongs to the domain vcs_domain, the syntax is *'VCS_DOMAIN\\John'=FULL_CONTROL* The domain name and the user name must be enclosed in quotation marks. Default is {'Everyone' = READ_ACCESS}. A maximum of 50 users can be configured for each file share. To configure more than 50 users for a file share, configure user groups. Note that the agent monitors only the users and the permissions that are defined in the VCS configuration. **Note:** This attribute can take localized values. Type and Dimension: string-association
AccessBasedEnumeration	Defines whether the agent enables the Windows Access-based Enumeration option for the specified file share. The value 1 indicates that the agent enables it and the value 0 indicates that the agent does not. Default is 0. Type and Dimension: boolean-scalar

Table 4-4 CompositeFileShare agent optional attributes *(continued)*

Optional Attributes	Description
ShareComment	A string that describes the share. Use this attribute to add comments to the compositefileshare resource. If this attribute is blank, the agent uses the default VCS comment. Type and Dimension: string-scalar

Sample configuration

```
CompositeCFS CFS-RES (
    Critical = 1
    LanmanResName = CFS_SG-Lanman
    MountResName = CFS_SG-MountV
    UserPermissions = { Everyone = READ_ONLY }
    ShareSubdirectories = 1
    PathAndShareName = { "\\share2" = share2,
    "\\share3" = share3 }
    AccessBasedEnumeration = 1
    )

IP CFS_SG-IP (
    Address = "10.212.100.184"
    SubNetMask = "255.255.254.0"
    MACAddress @SYSTEM1 = "00-02-B3-AD-CE-E4"
    MACAddress @SYSTEM2 = "00-02-B3-AD-CB-D0"
    )

Lanman CFS_SG-Lanman (
    VirtualName = VNAME
    IPResName = CFS_SG-IP
    )

MountV CFS_SG-MountV (
    MountPath = "J:"
    VolumeName = PS-SPOOL
    VMDGResName = CFS_SG-VMDg
    )

NIC CFS_SG-NIC (
    MACAddress @SYSTEM1 = "00-02-B3-AD-CE-E4"
```

```
    MACAddress @SYSTEM2 = "00-02-B3-AD-CB-D0"
    )

VMDg CFS_SG-VMDg (
    DiskGroupName = FILESHARE_DG
    DGGuid = e40b8ab5-88ac-42e3-ae61-cbea3a164af2
    )

CFS_SG-Lanman requires CFS_SG-IP
CFS_SG-MountV requires CFS_SG-VMDg
CFS-RES requires CFS_SG-Lanman
CFS-RES requires CFS_SG-MountV
CFS_SG-IP requires CFS_SG-NIC
```

Chapter 5

Services and applications agents

This chapter includes the following topics:

- About services and application agents
- Internet Information Services agent
- GenericService agent
- ServiceMonitor agent
- Process agent
- MSMQ agent
- AzureAuth agent

About services and application agents

Services and applications agents make generic services and other applications highly available.

- The Internet Information Services (IIS) agent brings IIS sites and services online, monitors their status, and takes them offline.
- The GenericService agent brings services online, monitors their status, and takes them offline.
- The ServiceMonitor agent only monitors a service or starts a user-defined script and interprets the exit code of the script.
- The Process agent brings processes online, monitors their status, and takes them offline.

- The MSMQ agent works with the Microsoft Message Queuing (MSMQ) service. The MSMQ agent creates an MSMQ service for an MSMQ resource, brings the MSMQ service online, takes it offline, and monitors its status.
- The AzureAuth agent authenticates the Azure subscription using service principal credentials.

Internet Information Services agent

The Internet Information Services (IIS) agent provides high availability to sites configured using Microsoft IIS. Specifically, the agent brings IIS sites online, monitors their status, and takes them offline.

The agent provides two ways of monitoring application pools associated with IIS Web sites:

- One IIS resource configures a Web site and sets monitoring options for application pools associated with the site.
- One IIS resource configures a Web site; other resources configure individual application pools.

The IIS agent supports intelligent resource monitoring and uses Intelligent Monitoring Framework (IMF) for resource state change notifications. IMF-based monitoring support is limited only to monitoring the IIS services (FTP service, World Wide Web Publishing Service) that are necessary for the functioning of IIS. The agent traps the Windows service related events and takes appropriate action if a configured service stops or fails to respond. IMF is not used for monitoring the availability of the sites configured.

For more information about IMF, refer to the *Cluster Server Administrator's Guide*.

Note: The IIS agent does not support IPv6 for IIS 6.0 or earlier versions.

Agent functions

online	Starts the configured site or application pool.
offline	Stops the configured site or application pool.
monitor	Verifies the configured site or application pool is running.

State definitions

ONLINE	Indicates the configured site or application pool is available.
OFFLINE	Indicates the configured site or application pool is not available.
UNKNOWN	Indicates the agent could not determine the status of the resource.

Resource type definition

```
type IIS (
    static i18nstr ArgList[] = {SiteType, SiteName,
    "IPResName:Address", PortNumber, AppPoolMon, DetailMonitor,
    DetailMonitorInterval }
    static int IMF{} = { Mode=2, MonitorFreq=1, RegisterRetryLimit=3 }
    str SiteType
    i18nstr SiteName
    int PortNumber = 80
    str AppPoolMon = NONE
    boolean DetailMonitor = 0
    int DetailMonitorInterval = 5
    str IPResName
)
```

Attribute definitions

To configure the agent to monitor an application pool, configure the SiteType and SiteName attributes only. The agent ignores other attributes when it is configured to monitor an application pool.

Table 5-1 IIS agent required attributes

Required Attributes	Description
SiteType	Defines whether the resource is configured to monitor an IIS site or an application pool. If the resource is configured to monitor an application pool, set the attribute to APPPOOL. If the resource is configured to monitor an IIS site, set this attribute to the name of the IIS service associated with the site. The attribute can take any of the following values: ■ W3SVC ■ MSFTPSVC ■ SMTPSVC Type and Dimension: string-scalar
SiteName	The name of the IIS site, the virtual server, or the application pool to be monitored by the agent. The value of this attribute depends on that of the SiteType attribute. The SiteName attribute can take the following values: ■ The name of a site, if SiteType is W3SVC or MSFTPSVC ■ The name of a virtual server, if SiteType is SMTPSVC ■ The name of an application pool, if SiteType is APPPOOL **Note:** This attribute can take localized values. Type and Dimension: string-scalar
IPResName	The name of the IP resource configured for the IP to which the site is bound. Type and Dimension: string-scalar
PortNumber	The port to which the site is bound. Type and Dimension: string-scalar

Table 5-2 IIS agent optional attributes

Optional Attributes	Description
AppPoolMon	Defines the monitoring modes for the application pool associated with the Web site being monitored. Configure this attribute only if SiteType is W3SVC and IIS is configured to run in the Worker Process Isolation mode. The attribute can take one of the following values: ■ NONE: Indicates that the agent will not monitor the application pool associated with the Web site. ■ DEFAULT: Indicates that the agent will monitor the *root* application pool associated with the Web site. If this attribute is set, the agent starts, stops, and monitors the root application pool associated with the Web site. If the root application pool is stopped externally, the agent fails over the service group. ■ ALL: Indicates the agent will start all application pools associated with the Web site, but will monitor and stop the *root* application pool only. Type and Dimension: integer-scalar
DetailMonitor	Defines whether the agent monitors the site in detail. The value 1 indicates the agent will monitor each site in detail by attempting an actual socket connection to the port. Type and Dimension: boolean-scalar
DetailMonitorInterval	The number of monitor cycles after which the agent attempts detail monitoring. For example, the value 5 indicates that the agent will monitor the resource in detail after every 5 monitor cycles. Type and Dimension: integer-scalar

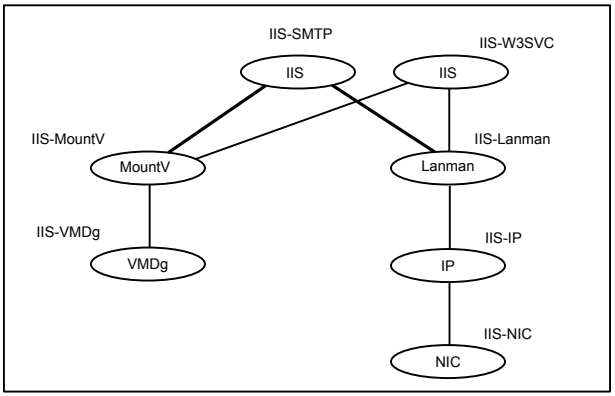
Sample configurations

This section describes two sample configurations. The first configuration depicts how you can configure one IIS resource to monitor a Web site and its associated application pools. The second configuration depicts how you can configure separate IIS resources to monitor Web sites and application pools.

Sample configuration I

In the following configuration the IIS sites are located in a shared disk group, which is configured using the VMDg resource type.

Figure 5-1 IIS service group dependency graph



A MountV resource configures a drive letter for the shared disk group. The IIS sites share a virtual name and IP address, configured using the Lanman and IP resource types respectively. The IIS resources depend on the Lanman and MountV resources and can be started after these resources are brought online.

Note: In case of a NetApp storage environment, the NetApp Filer and NetApp SnapDrive agents are used instead of the VMDg and MountV agents respectively.

Sample main.cf file

```
IIS IIS-W3SVC (
    SiteType = W3SVC
    SiteName = "My Site"
    AppPoolMon = DEFAULT
    DetailMonitor = 1
    IPResName = IIS-IP
)

IIS IIS-SMTP (
    SiteType = SMTPSVC
    SiteName = "Default SMTP Virtual Server"
    PortNumber = 25
    DetailMonitor = 1
    IPResName = IIS-IP
)


IP IIS-IP (
```

```
    Address = "10.182.145.223"
    SubNetMask = "255.255.252.0"
    MACAddress @SYSTEM1 = "00-50-04-1E-62-E0"
    MACAddress @SYSTEM2 = "00-50-04-25-0E-C0"
)

Lanman IIS-Lanman (
    VirtualName = VNAME
    IPResName = IIS-IP
)

MountV IIS-MountV (
    MountPath = "X:"
    VolumeName = web
    VMDGResName = IIS-VMDg
    ForceUnmount = ALL
)

NIC IIS-NIC (
    MACAddress @SYSTEM1 = "00-50-04-1E-62-E0"
    MACAddress @SYSTEM2 = "00-50-04-25-0E-C0"
)

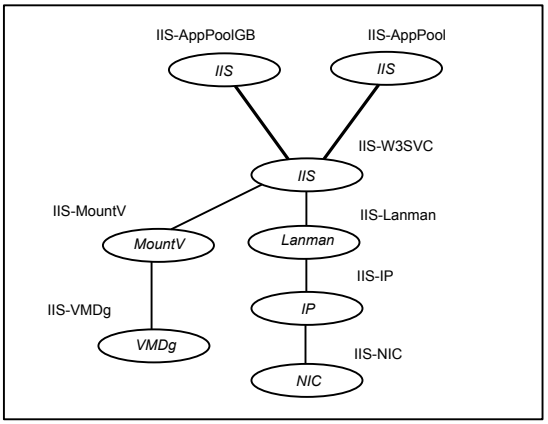
VMDg IIS-VMDg (
    DiskGroupName = DG_1
)

IIS-W3SVC requires IIS-Lanman
IIS-W3SVC requires IIS-MountV
IIS-SMTP requires IIS-Lanman
IIS-SMTP requires IIS-MountV
IIS-IP requires IIS-NIC
IIS-Lanman requires IIS-IP
IIS-MountV requires IIS-VMDg
```

Sample configuration II

In the following configuration the IIS Web site is located in a shared disk group, which is configured using the VMDg resource type.

Figure 5-2 IIS service group dependency graph



A MountV resource configures a drive letter for the shared disk group. The virtual name and IP address for the IIS resource are configured using the Lanman and IP resource types respectively. The application pools associated with the Web site are configured as separate resources of type IIS.

Note: In case of a NetApp storage environment, the NetApp Filer and NetApp SnapDrive agents are used instead of the VMDg and MountV agents respectively.

Sample main.cf file

```
IIS IIS-W3SVC (
    SiteType = W3SVC
    SiteName = "My Site"
    AppPoolMon = DEFAULT
    DetailMonitor = 1
    IPResName = IIS-IP
)

IIS IIS-APPPOOL (
    SiteType = APPPOOL
    SiteName = DefaultAppPool
)

IIS IIS-APPPOOL_GB (
    SiteType = APPPOOL
    SiteName = GuestBook
)
```

```
IP IIS-IP (
    Address = "10.182.145.223"
    SubNetMask = "255.255.252.0"
    MACAddress @SYSTEM1 = "00-50-04-1E-62-E0"
    MACAddress @SYSTEM2 = "00-50-04-25-0E-C0"
)

Lanman IIS-Lanman (
    VirtualName = VNAME
    IPResName = IIS-IP
)

MountV IIS-MountV (
    MountPath = "X:"
    VolumeName = web
    VMDGResName = IIS-VMDg
    ForceUnmount = ALL
)

NIC IIS-NIC (
    MACAddress @SYSTEM1 = "00-50-04-1E-62-E0"
    MACAddress @SYSTEM2 = "00-50-04-25-0E-C0"
)

VMDg IIS-VMDg (
    DiskGroupName = DG_1
)
IIS-W3SVC requires IIS-Lanman
IIS-W3SVC requires IIS-MountV
IIS-IP requires IIS-NIC
IIS-Lanman requires IIS-IP
IIS-MountV requires IIS-VMDg
IIS-APPPOOL requires IIS-W3SVC
IIS-APPPOOL_GB requires IIS-W3SVC
```

GenericService agent

The GenericService agent brings services online, takes them offline, and monitors their status. Note that a service is an application type supported by Windows that conforms to the interface rules of the Service Control Manager (SCM).

Services are defined as resources of type GenericService. You can configure the GenericService agent to monitor multiple services by defining a resource for each service to be monitored. You can monitor a service in a user-context by specifying the user name, password, and domain. You can start and monitor services in the virtual computer context by enabling the UseVirtualName attribute.

Configure a service group for GenericService agent using the Application Configuration wizard, described in the *Cluster Server Administrator's Guide*.

This agent supports intelligent resource monitoring and uses Intelligent Monitoring Framework (IMF) for resource state change notifications. The agent traps the Windows service related events and takes appropriate action if a configured service stops or fails to respond.

For more information about IMF, refer to the *Cluster Server Administrator's Guide*.

Note: The service to be configured using the GenericService agent must have the status as Stopped and the startup type as Manual.

Agent functions

online	Starts the configured service.
offline	Stops the configured service.
monitor	Retrieves the current state of the configured service. It also verifies the user context, if applicable.

State definitions

`ONLINE`	Indicates the service being monitored is online.
`OFFLINE`	Indicates the service being monitored is offline.
`UNKNOWN`	Indicates the service operation is in a pending state, or that the agent could not determine the state of the resource.

Resource type definition

```
type GenericService (
    static i18nstr ArgList[] = {ServiceName, DelayAfterOnline,
    DelayAfterOffline, UserAccount, Password, Domain, service_arg,
    UserVirtualName, "LanmanResName:VirtualName" }
    static int IMF{} = { Mode=3, MonitorFreq=5, RegisterRetryLimit=3 }
```

```
    i18nstr ServiceName
    int DelayAfterOnline = 10
    int DelayAfterOffline = 10
    i18nstr UserAccount
    str Password
    i18nstr Domain
    str service_arg[]
    boolean UseVirtualName
    str LanmanResName
)
```

Attribute definitions

Table 5-3 GenericService agent required attributes

Required Attribute	Description
ServiceName	Name of the service to be monitored. The service name can be the Service Display Name or the Service Key Name. **Note:** This attribute can take localized values. Type and Dimension: string-scalar

Table 5-4 GenericService agent optional attributes

Optional Attributes	Description
DelayAfterOffline	Number of seconds the offline routine waits for the service to go offline. Modify this value depending on the time the configured service takes to stop once the resource is taken offline in the environment. Default is 10 seconds. Type and Dimension: integer-scalar
DelayAfterOnline	Number of seconds the online routine waits for the service to go online. Modify this value depending on the time the configured service takes to start once the resource is taken online in the environment. Default is 10 seconds. Type and Dimension: integer-scalar

Table 5-4 GenericService agent optional attributes *(continued)*

Optional Attributes	Description
Domain	The domain to which the user specified in the UserAccount attribute belongs. If the domain name is not specified, the agent assumes that the user belongs to the local machine. This attribute is ignored if the UserAccount attribute is not specified or if the domain name is specified in the UserAccount attribute itself. **Note:** This attribute can take localized values. Type and Dimension: string-scalar
Password	The password of the user, in whose context, the service would be started. If the UserAccount attribute is empty or contains a built-in service account, this attribute is ignored. Type and Dimension: string-scalar
service_arg	An array of arguments passed to the service. Type and Dimension: string-vector
UserAccount	A valid user account in whose context the service will be monitored. Username can be of the form username@domain.com or domain.com\username. If you do not specify a value for this attribute, then the user account of the service in the SCM is ignored. To monitor service under built-in accounts, you must provide explicit values. For example: UserAccount='LocalSystem', 'Local Service', or 'Network Service'. Domain='NT Authority'. The 'NT Authority' domain is not applicable for the 'LocalSystem' account. **Note:** This attribute can take localized values. Type and Dimension: string-scalar

Table 5-4 GenericService agent optional attributes *(continued)*

Optional Attributes	Description
UseVirtualName	Set this value to enable the service specified by the ServiceName attribute to inherit the Virtual Server context of Lanman resource specified in LanmanResName attribute. If this attribute is set, the LanmanResName becomes a mandatory attribute. **Note:** It is recommended that you do not set this attribute for system services. If you configure system services using the GenericService agent, then this attribute is not supported. Type and Dimension: boolean-scalar
LanmanResName	The name of the Lanman resource on which the GenericService resource depends. **Note:** It is recommended that you do not set this attribute for system services. If you configure system services using the GenericService agent, then this attribute is not supported. Type and Dimension: string-scalar

Sample configuration

```
GenericService Telephony_Service (
    ServiceName = "Telephony Service"
    DelayAfterOnline = 10
    DelayAfterOffline = 10
    UserAccount = Administrator
    Password = bphNepE
    Domain = VCS-DOMAIN
    service_arg = { 3 }
)
```

ServiceMonitor agent

The ServiceMonitor agent monitors a service or starts a user-defined script and interprets the exit code of the script. A service is an application type supported by Windows that conforms to the interface rules of the SCM. The agent can be configured to monitor a service using the Service Control Manager (SCM) or through a user-defined script.

When configured to monitor using the SCM, the agent queries the SCM database for the status of the service. When configured to monitor using a script or an

executable, the agent executes the script on each system in the cluster, once every monitor cycle. The agent returns `ONLINE`, `OFFLINE`, or `UNKNOWN` depending on the exit code of the script. The exit code must conform to VCS standards.

Refer to chapter on agent functions in the *Cluster Server Agent Developer's Guide* for details.

The ServiceMonitor agent supports intelligent resource monitoring and uses Intelligent Monitoring Framework (IMF) for resource state change notifications. The agent traps the Windows service related events and takes appropriate action if a configured service stops or fails to respond.

For more information about IMF, refer to the *Cluster Server Administrator's Guide*.

You can monitor a service or run a script in a user-context by specifying the user name, password, and domain. Configure a service group for ServiceMonitor agent using the Application Configuration wizard.

The Application Configuration wizard is described in the *Cluster Server Administrator's Guide*.

Note: The ServiceMonitor resource cannot be made dependent on other resources.

Agent functions

online	Not applicable.
offline	Not applicable.
monitor	If the agent is configured to monitor a service, the agent queries the SCM for the status of the service. If the agent is configured to start a script, the agent spawns the specified script and interprets the exit code of the script.

State definitions

`ONLINE`	Indicates the specified service is running, or that the configured script returned an online exit code.
`FAULTED`	Indicates the specified service is not running, or that the configured script returned an offline exit code.
`UNKNOWN`	Indicates the agent could not determine the state of the resource.

Resource type definition

```
type ServiceMonitor (
    static i18nstr ArgList[] = {MonitorService, ServiceOrScriptName,
    MonitorProgTimeout, UserAccount, Password, Domain }
    static int IMF{} = { Mode=3, MonitorFreq=5, RegisterRetryLimit=3 }
    boolean MonitorService = 1
    i18nstr ServiceOrScriptName
    int MonitorProgTimeout = 30
    i18nstr UserAccount
    str Password
    i18nstr Domain
)
```

Attribute definitions

Table 5-5 ServiceMonitor agent required attributes

Required Attribute	Description
ServiceOrScriptName	The name of the service to be monitored or script to be spawned for monitoring the service. You must specify the complete path of the script, including any command-line arguments. This attribute can take localized values. **Note:** The exit code of the script must conform to VCS conventions: 110 for ONLINE and 100 for OFFLINE. For exit values outside the range 100-110, the status is considered `UNKNOWN`. Refer to the *Cluster Server Agent Developer's Guide* for more information. Type and Dimension: string-scalar

Table 5-6 ServiceMonitor agent optional attributes

Optional Attributes	Description
Domain	The domain to which the user belongs, as specified by the attribute UserAccount. If the domain name is not specified, the agent assumes the user to belong to the local machine. This attribute is ignored if the UserAccount attribute is not specified or if the domain name is specified in the UserAccount attribute itself. **Note:** This attribute can take localized values. Type and Dimension: string-scalar
MonitorProgTimeout	The maximum wait time, in seconds, for the agent to receive a return value from the monitor script. This attribute is ignored if the MonitorService attribute is set to 1. Default is 30 seconds. Type and Dimension: integer-scalar
MonitorService	Defines whether the agent monitors a service or starts a script. If set to 1, the agent monitors a service specified by the attribute ServiceOrScriptName. If set to 0 the agent starts a script specified by the attribute ServiceOrScriptName. Default is 1. Type and Dimension: boolean-scalar
Password	The password for the user account, encrypted using the VCS Encrypt utility. This attribute is valid only if the MonitorService is set to 0 and UserAccount is not empty. For more information about the VCS Encrypt utility, see the *Cluster Server Administrator's Guide*. Type and Dimension: string-scalar

Table 5-6 ServiceMonitor agent optional attributes *(continued)*

Optional Attributes	Description
UserAccount	A valid user account in whose context the service will be monitored. Username can be of the form *username@domain.com* or *domain.com\username*. If you do not specify a value for this attribute, then the user account of the service in the SCM is ignored. To monitor service under built-in accounts, you must provide explicit values. For example: User Account="LocalSystem", "Local Service", or "Network Service". Domain="NT Authority". The 'NT Authority' domain is not applicable for the 'LocalSystem' account. **Note:** This attribute can take localized values. Type and Dimension: string-scalar

Sample configurations

In the following example, the ServiceMonitor agent is configured to spawn a script and interpret the script's exit code.

```
ServiceMonitor ScriptBasedMonitor (
    MonitorService = 0
    ServiceOrScriptName = "C:\\Programs\\monitor-script.bat"
    MonitorProgTimeout = 15
    )
```

In the next example, the agent is configured to monitor a service.

```
ServiceMonitor Telephony_Service (
    MonitorService = 1
    ServiceOrScriptName = "Telephony Service"
    UserAccount = Administrator
    Password = bphNepE
    Domain = VCS-DOMAIN7
)
```

Note: While monitoring a service, the Password attribute is ignored.

Process agent

The Process agent brings processes online, takes them offline, and monitors their status. You can specify different executables for each process routine. You can configure the Process agent to monitor processes in a virtual server context. By default, processes are monitored in the context of the LocalSystem account. You can run a process with user privileges by specifying the user name, password, and domain.

You can configure a service group for the Process agent using the Application Configuration wizard or manually using the command line.

Refer to the *Cluster Server Administrator's Guide* for more details.

The Process agent supports intelligent resource monitoring and uses Intelligent Monitoring Framework (IMF) for resource state change notifications. The agent supports IMF-based monitoring only when the resource is in the online state.

For more information about IMF, refer to the *Cluster Server Administrator's Guide*.

Note: The Process agent does not use IMF notification for monitoring the program specified in the MonitorProgram attribute.

Agent functions

online	Starts the process configured as the start program.
offline	Terminates the process, or starts the process configured as the stop program.
monitor	Verifies the status of the process, or starts the process configured as the monitor program.

State definitions

`ONLINE`	Indicates the process being monitored is running properly.
`OFFLINE`	Indicates the process being monitored is not running properly.
`UNKNOWN`	Indicates the agent could not determine the status of the resource.

Resource type definition

```
type Process (
    static i18nstr ArgList[] = { StartProgram, StopProgram,
    MonitorProgram, UserName, Password, Domain,
    MonitorProgramTimeout,InteractWithDesktop, CleanProgram,
    StartupDirectory, StopProgramTimeout, CleanProgramTimeout,
    "LanmanResName:VirtualName" }
    static int IMF{} = { Mode=2, MonitorFreq=5, RegisterRetryLimit=3 }
    i18nstr StartProgram
    i18nstr StartupDirectory
    i18nstr StopProgram
    i18nstr CleanProgram
    i18nstr MonitorProgram
    i18nstr UserName
    str Password
    i18nstr Domain
    int MonitorProgramTimeout = 30
    boolean InteractWithDesktop = 0
    int StopProgramTimeout = 30
    int CleanProgramTimeout = 30
    str LanmanResName
)
```

Attribute definitions

Table 5-7 Process agent required attributes

Required Attribute	Description
StartProgram	The process to be monitored by the agent. You must specify the complete path of the executable, its file extension, and command-line arguments, if any. If you define the start program as a batch file or a script to launch another program, you must specify the monitor program in the configuration file. If you define the start program as a script (a perl script, or a vbs script), the start program should be the program that interprets the script (perl.exe, or cscript.exe) and the script itself should be passed as an argument. **Note:** This attribute can take localized values. Type and Dimension: string-scalar

Table 5-8 Process agent optional attributes

Optional Attributes	Description
CleanProgram	The full path of the clean process that is launched when the resource needs a forceful offline. If no value is specified for this attribute, for a clean operation the agent kills the process indicated by the StartProgram attribute. **Note:** This attribute can take localized values. Type and Dimension: string-scalar
CleanProgramTimeout	The maximum time, in seconds, that the agent must wait before killing the process specified in the CleanProgram attribute. The default value is 30 seconds. Type and Dimension: integer-scalar
Domain	The domain to which the user specified in the UserName attribute belongs. If the domain name is not specified, the agent assumes that the user belongs to the local machine. This attribute is ignored if the UserName attribute is not specified or if the domain name is specified in the UserName attribute itself. **Note:** This attribute can take localized values. Type and Dimension: string-scalar
InteractWithDesktop	Defines whether the configured process interacts with the desktop. Enabling desktop interaction enables user intervention for the process. The value 1 indicates the process will interact with the desktop. The value 0 indicates it will not. Default is 0. Type and Dimension: boolean-scalar

Table 5-8 Process agent optional attributes *(continued)*

Optional Attributes	Description
MonitorProgram	A program that monitors the process specified as the start program. You must specify the complete path of the executable, its file extension, and command-line arguments, if any. If you do not specify a value for this attribute, VCS monitors the start program. However, if the start program is a batch file or a script to launch another program, you must specify a monitor program. The Process agent supports Intelligent Monitoring Framework (IMF). However, IMF is not supported for the MonitorProgram attribute. If a script or a batch file is specified for this attribute, the Process agent does not use IMF notification to monitor the program. The agent detects the faults only during the regular monitor function. This attribute can take localized values. **Note:** The monitor program is spawned every monitor cycle and must return before the program specified in MonitorProgram times out. The return values for the monitor program must conform to VCS conventions: 110 for `ONLINE` and 100 for `OFFLINE`. For exit values outside the range 100-110, the status is considered `UNKNOWN`. Refer to the *Cluster Server Agent Developer's Guide* for more information. Type and Dimension: string-scalar
MonitorProgramTimeout	The maximum wait time, in seconds, for the agent to receive a return value from the monitor routine. This attribute is ignored if the monitor program is not specified. Default is 30 seconds. Type and Dimension: integer-scalar
Password	The encrypted password of the user specified by the UserName. Note that the password must be encrypted using the VCSEncrypt utility. See the *Cluster Server Administrator's Guide* for more information about the utility. This attribute is ignored if the user name is not specified. Type and Dimension: string-scalar
StartupDirectory	The startup directory for the process indicated by the StartProgram attribute. You must enter the startup directory name in double quotes. **Note:** This attribute can take localized values. Type and Dimension: string-scalar

Table 5-8 Process agent optional attributes *(continued)*

Optional Attributes	Description
StopProgram	A program that stops the process specified as the start program. You must specify the complete path of the program, its file extension, and command-line arguments, if any. If you do not specify a value for this attribute, VCS stops the start program. This attribute can take localized values. **Note:** If successful, the StopProgram returns a positive value. The agent framework calls the Monitor routine after those many seconds, as returned by StopProgram. Also, while writing a stop program, make sure to stop all the processes launched by the start program. Type and Dimension: string-scalar
StopProgramTimeout	The maximum time, in seconds, that the agent must wait before killing the process specified in the StopProgram attribute. The default value is 30 seconds. Type and Dimension: integer-scalar
UserName	The user name in whose context the configured process executes. Username can be of the form username@domain.com or domain.com\username. If a user name is not specified, the configured process runs in the context of the user account used by VCS High Availability Engine (HAD). **Note:** This attribute can take localized values. Type and Dimension: string-scalar
LanmanResName	The name of the Lanman resource. The Process agent monitors the configured process in the context of the virtual name specified in the Lanman resource. Type and Dimension: string-scalar

Note: When defining the StartProgram, StopProgram, CleanProgram, or MonitorProgram attributes, enclose the path of the executable file in double quotes. Do not enclose arguments in double quotes.

Sample configuration

Sample 1

In the following configuration, the online function spawns the executable test.exe with `online` as a command-line argument. The `offline` function spawns the executable test.exe with `offline` as the command-line argument. The `monitor` function monitors the attribute value for StartProgram, test.exe, and checks the state.

```
Process Process_Test (
    StartProgram = "\"X:\\bin\\test.exe\" /online"
    StopProgram = "\"X:\\bin\\test.exe\" /offline"
    UserName = Administrator
    Password = bphNepE
    Domain = "VCS_DOMAIN"
)
```

Sample 2

In the following configuration, the batch file start.bat starts an instance, `MyProcess`, of the executable process.exe. The batch file stop.bat stops the process instance of process.exe. The batch file monitor.bat monitors the attribute value for StartProgram, and checks the state of the process.exe. Note that the executable process.exe resides in the directory `C:\batchfiles`, which is specified as the StartupDirectory.

To configure the process agent by editing the configuration file main.cf directly, the configuration is as follows:

```
PROCESS-Process_SG (
    StartProgram = "\"C:\\batchfiles\\start.bat\" MyProcess"
    StartupDirectory = "C:\\batchfiles"
    StopProgram = "\"C:\\batchfiles\\stop.bat\" MyProcess"
    MonitorProgram = "\"C:\\batchfiles\\monitor.bat\" MyProcess"
    UserName = Administrator
    Password = jxpVmxM
    Domain = "mydomain.mycompany.com"
    )
```

To configure the Process agent using the Cluster Manager, enter the attributes as follows:

```
StartProgram = "C:\batchfiles\start.bat" MyProcess
StartupDirectory = "C:\batchfiles"
StopProgram = "C:\batchfiles\stop.bat" MyProcess
```

```
MonitorProgram = "C:\batchfiles\monitor.bat" MyProcess
UserName = Administrator
Password = jxpVmxM
Domain = "mydomain.mycompany.com"
```

MSMQ agent

The MSMQ agent is a bundled agent that works with the Microsoft Message Queuing (MSMQ) service. The MSMQ agent creates an MSMQ service for an MSMQ resource, brings the service online, takes it offline, and monitors its status. The agent provides high availability for the MSMQ service and the queues hosted by the service, in a clustered environment.

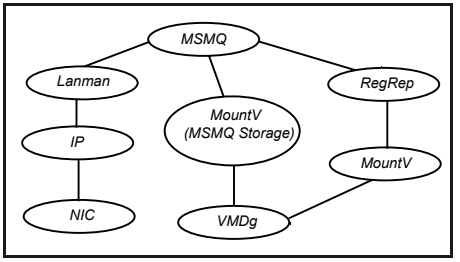
You can configure an MSMQ resource using the command line. See the *Cluster Server Administrator's Guide* for instructions.

Dependencies

The MSMQ resource depends on the MountV, Lanman, and the Registry Replication resources for operation. The MSMQ resource starts after each of these resources is brought online.

The following figure illustrates the agent's dependency graph.

Figure 5-3 MSMQ agent dependency graph



Note: In case of a NetApp storage environment, the NetApp Filer and NetApp SnapDrive agents are used instead of the VMDg and MountV agents respectively.

Agent functions

online	Starts the MSMQ service.
offline	Stops the MSMQ service.

monitor	Verifies that the MSMQ service is running.

State definitions

`ONLINE`	Indicates that the MSMQ service is running properly.
`OFFLINE`	Indicates that the MSMQ service has stopped.
`UNKNOWN`	Indicates that the agent could not determine the status of the MSMQ service.

Resource type definition

```
type MSMQ (
    static i18nstr ArgList[] ={"IPResName:Address",
    "LanmanResName:VirtualName","MountResName:MountPath",
    StoragePath, DetailMonitor }
    str IPResName
    str LanmanResName
    str MountResName
    i18nstr StoragePath
    boolean DetailMonitor = 0
)
```

Attribute definitions

Table 5-9 MSMQ agent required attributes

Required Attribute	Description
IPResName	The name of the IP resource. Configure the IP resource to provide a virtual IP binding for the MSMQ service. **Note:** Every time you make changes to this attribute, you must disable the MSMQ resource and enable it again. Type and Dimension: string-scalar
LanmanResName	The name of the Lanman resource. MSMQ uses the virtual name configured in the specified Lanman resource to set the virtual environment for the MSMQ service. Type and Dimension: string-scalar

Table 5-9 MSMQ agent required attributes *(continued)*

Required Attribute	Description
MountResName	The name of the MountV resource. Configure the MountV resource to monitor the volume on which the MSMQ service stores its queue related data. In case of a NetApp storage environment, this will be the name of the NetAppSnapDrive resource. Type and Dimension: string-scalar
StoragePath	The root storage path of the MSMQ service. **Note:** This attribute can take localized values. Type and Dimension: string-scalar
DetailMonitor	Used to enable detail monitoring for the MSMQ service. Type and Dimension: boolean-scalar

Sample configuration

The following configuration creates the MSMQ resource with the required Lanman, MountV, and the RegRep resources.

```
MSMQ test (
    IPResName=msmqip
    LanmanResName=msmqlanman
    MountResName=msmqMountV
    StoragePath=\\msmqStorage
)
```

AzureAuth agent

To perform any operation on Azure resources, such as updating a resource record set, attaching an Azure data disk, assigning a private IP to a Network Interface, and so on requires you to authenticate that you are an authorized Azure user.

AzureAuth agent authenticates the Azure subscription using service principal credentials.

AzureAuth agent is a persistent resource that monitors the validity of service principal credentials.

Prerequisites

- Create the service principal from the Azure portal and assign the application to a role. For details, refer to the Microsoft Azure documentation.
- Ensure that the credentials that are passed on to the AzureAuth agent have at least the minimum required role assigned to service principal.
  The minimum roles required for each agent are:
  - AzureIP: Network Contributor and Virtual Machine Contributor
  - AzureDisk
    - Un-Managed Disks: Virtual Machine Contributor
    - Managed Disks: Contributor
  - AzureDNSZone: DNS Zone Contributor
- Obtain the authentication keys (SubscriptionId, ClientId, SecretKey, and TenantId).
  See "To obtain the Azure authentication keys" on page 154.
- Install Python SDK for Azure on all cluster nodes.
  Python SDK can be installed with `pip` as follows:

```
C:\Program Files\Veritas\VRTSPython>python -m pip install
azure.mgmt.compute==3.0.0
C:\Program Files\Veritas\VRTSPython>python -m pip install
azure-mgmt-network==1.7.1
C:\Program Files\Veritas\VRTSPython>python -m pip install
azure.mgmt.dns==1.0.0
C:\Program Files\Veritas\VRTSPython>python -m pip install
azure.storage==0.33.0
C:\Program Files\Veritas\VRTSPython>python -m pip install
--force-reinstall msrest==0.4.29
C:\Program Files\Veritas\VRTSPython>python -m pip install
--force-reinstall msrestazure==0.4.19
C:\Program Files\Veritas\VRTSPython>python -m pip install
--force-reinstall azure-common==1.1.4
```

Dependency

The AzureAuth agent is not dependent on any other resources.

Agent functions

Monitor	Validates the service principal credentials with Azure.

State definitions

ONLINE — Indicates that the service principal credentials are valid.

UNKNOWN — Indicates that one of the following is true:

- The service principal credentials are invalid
- The service principal credentials were modified or deleted after the resource reported ONLINE
- The service principal credentials are expired

Attributes

Table 5-10 Required attributes

Attribute	Description
SubscriptionId	Identifier that uniquely identifies your Azure subscription. Type and dimension: string-scalar
ClientId	Identifier of the Azure Active Directory (AAD) Application. Type and dimension: string-scalar
SecretKey	Authentication key generated for the AAD application. You must encrypt this secret key using the `vcsencrypt -agent` command. Type and dimension: string-scalar
TenantId	Identifier of the AAD directory in which you created the application. Type and dimension: string-scalar

Resource type definition

```
type AzureAuth (
    static i18nstr ArgList[] = { SubscriptionId, ClientId, SecretKey,
      TenantId }
    static str Operations = None
    str SubscriptionId
    str ClientId
    str SecretKey
    str TenantId
)
```

Sample configuration

```
AzureAuth Auth_Res (
    SubscriptionId = 2dfgg136-fgh6-40dd-b616-c1e9abdf1d63
```

```
    ClientId = 123456-d10a-4704-8986-beb86739104d
    SecretKey = fntPgnUnhTprQrqrnRonSlhPhrQpiNtrItpRhnGrrNklFngLs
    TenantId = 12345-0528-4308-brf03-6667d61dd0e3
)
```

Obtaining authentication keys

To obtain the Azure authentication keys

1. Log in to the Azure portal.
2. Perform the following to obtain the authentication keys :

Task	Procedure
To obtain the SubscriptionId	**1** In the left navigation pane, click **Subscriptions**. A list of your subscriptions is displayed along with the subscription IDs. **2** Copy and provide this Id as SubscriptionId to the AzureAuth agent.
To obtain the TenantID	**1** In the left navigation panel, click **Azure Active Directory**. **2** On the page that opens, click **Properties**. **3** Copy the **Directory ID**. **4** Provided this Id as TenantID to the AzureAuth agent.
To obtain the ClientId	**1** In the left navigation panel, click **Azure Active Directory**. **2** On the page that opens, click **App registrations**. **3** Search and select your application (service principal) from the list of applications. **4** Copy the **Application ID**. **5** Provide this Id as the ClientId to the AzureAuth agent.

Task	Procedure
To obtain the SecretKey	**1** From **App registrations** in **Azure Active Directory**, search and select your application. **2** Under **Settings**, select **Keys**. **3** Provide a description of the key, and the expiry duration for the key. **4** Click **Save**. The key is displayed. **Note:** Ensure that you copy and store the key value. You cannot retrieve the key later. **5** Encrypt the key using: `vcsencrypt -agent < Secret Key>`. **6** Provide the encrypted key to the AzureAuth agent as the SecretKey.

Chapter 6

VCS infrastructure and support agents

This chapter includes the following topics:

- About VCS infrastructure and support agents
- NotifierMngr agent
- Registry Replication (RegRep) agent
- Proxy agent
- Phantom agent
- FileNone agent
- ElifNone agent
- FileOnOff agent
- FileOnOnly agent
- RemoteGroup agent

About VCS infrastructure and support agents

The VCS infrastructure and support agents provide high availability for VCS-related operations. You can use some of these agents to test VCS functionality.

- NotifierMngr agent—Monitors a notifier process.
- RegRep Agent—Replicates the system registry.
- Proxy Agent—Monitors the state of a resource on a local or remote system.

- Phantom Agent—Enables VCS to determine the status of parallel service groups that do not include OnOff resources.
- ElifNone Agent—Monitors a file.
- FileNone Agent—Monitors a file.
- FileOnOff Agent—Monitors a file.
- FileOnOnly Agent—Monitors a file.
- RemoteGroup Agent—Monitors and manages service groups in a remote cluster.

NotifierMngr agent

The NotifierMngr agent starts, stops, and monitors a notifier process, making it highly available. The notifier process manages the reception of messages from VCS and Global Cluster Option (GCO) and the delivery of the messages to SNMP consoles and SMTP servers. VCS can send notification to SNMP console and SMTP servers.

InfoScale supports the NotifierMngr agent in IPv4, IPv6, and dual-stack configurations. In a dual-stack configuration, the agent can communicate with the SNMP and the SMTP servers only if both the servers have both IPv4 and IPv6 enabled.

See the *Cluster Server Administrator's Guide* for a description of types of events that generate notification.

The agent is represented by the NotifierMngr resource type. Note that only one instance of the notifier process can run in a cluster. Therefore, a notifier process cannot run in a parallel service group.

Note: The attributes of the NotifierMngr agent cannot be changed dynamically using the `hares -modify` command. Changes made using this command are effective after notifier is restarted.

Agent functions

online	Starts the notifier process with its required arguments.
offline	Stops the notifier process.
monitor	Monitors the notifier process.

State definitions

ONLINE	Indicates the notifier process is running.
OFFLINE	Indicates the notifier process is not running.
UNKNOWN	Indicates the agent encountered errors while monitoring the NotifierMgr resource.

Resource type definition

```
type NotifierMngr (
    static i18nstr ArgList[] = {MessageQueue, NotifierListeningPort,
    SnmpdTrapPort, SnmpCommunity, SnmpConsoles, SmtpServer,
    SmtpServerVrfyOff, SmtpServerTimeout, SmtpReturnPath,
    SmtpFromPath, SmtpRecipients }
    int MessageQueue = 30
    int NotifierListeningPort = 14144
    int SnmpdTrapPort = 162
    str SnmpCommunity = public
    str SnmpConsoles{}
    str SmtpServer
    int SmtpServerTimeout
    str SmtpReturnPath
    str SmtpFromPath
    str SmtpRecipients{}
    boolean SmtpServerVrfyOff
)
```

Attribute definitions

Table 6-1 NotifierMngr agent required attributes

Required Attributes	Description
SnmpConsoles	The system name or IPv4 address of the SNMP manager and the severity level of the messages to be delivered to the SNMP manager. The severity levels of messages are Information, Warning (default), Error, and SevereError. Specifying a severity level for messages generates delivery of all messages of equal or higher severity. For example: "172.29.10.89"=Error "172.29.10.56"=Information Type and Dimension: string-association
SmtpServer	The system name of the SMTP server. Type and Dimension: string-scalar

Table 6-2 NotifierMngr agent optional attributes

Optional Attributes	Description
MessageQueue	Size of the VCS engine's message queue size. Default is 30 messages. Type and Dimension: integer-scalar
NotifierListeningPort	Any valid, unused TCP/IP port number. Default is 14144. Type and Dimension: integer-scalar
SnmpdTrapPort	The port on which SNMP trap PDUs are sent. The value specified for this attribute is used for all consoles if more than one SNMP console is specified. Default is 162. Type and Dimension: integer-scalar
SnmpCommunity	The community ID for the SNMP manager. Default is "public." Type and Dimension: string-scalar
SmtpFromPath	This attribute must be set to a valid email address, if a custom email address is desired for the FROM: field in the email sent by notifier. Type and Dimension:string

Table 6-2 NotifierMngr agent optional attributes *(continued)*

Optional Attributes	Description
SmtpRecipients	The email address to which information is sent and the severity level of the messages to be received. The severity levels of messages are Information, Warning (default), Error, and SevereError. Specifying a severity level for messages indicates that all messages of equal or higher severity are received. Example: `"james@veritas.com"=SevereError "admin@veritas.com"=Warning` **Note:** SmtpReceipents is a required attribute if SmtpServer is specified. Type and Dimension: string-association
SmtpReturnPath	This attribute must be set to a valid email address, if a custom email address is desired for the Return-Path: field in the email sent by notifier. **Note:** If the mail server specified in the SmtpServer does not support VRFY, then you need to set the SmtpServerVrfyOff to 1 in order for the SmtpReturnPath to come into effect. Type and Dimension: string
SmtpServerTimeout	This attribute represents the time, in seconds, for which the notifier waits for a response from the mail server for the SMTP commands sent to the mail server. Increase the specified value if the mail server takes long to send response for the SMTP commands sent by the notifier. Default is 10. Type and Dimension: integer
SmtpServerVrfyOff	Setting this value to 1 results in notifier not sending a SMTP VRFY request to the mail server specified in SmptServer attribute, while sending emails. Set this to 1 if your mail server does not support SMTP VRFY command. Default is 0. Type and Dimension: boolean

Sample configuration

The NotifierMngr resource sets up notification for all events to the SnmpConsole, your_console. In the following example, only messages of SevereError level are sent to the SmptServer, "smtp.your_company.com," and the recipient, "vcsadmin@your_company.com."

Note: As only one instance of the notifier process can run in a cluster, the process cannot run in a parallel group.

```
IP csg_ip (
    Address = "10.212.100.14"
    SubNetMask = "255.255.254.0"
    MACAddress @SYSTEM1 = "00:B0:D0:D1:69:7D"
    MACAddress @SYSTEM2 = "00:B0:D0:AB:4E:22"
    )

NIC csg_nic (
    MACAddress @SYSTEM1 = "00:B0:D0:D1:69:7D"
    MACAddress @SYSTEM2 = "00:B0:D0:AB:4E:22"
    )

NotifierMngr ntfr(
    SnmpConsoles = { your_console = Information }
    SmtpServer   = "smtp.your_company.com"
    SmtpRecipients = { "vcsadmin@your_company.com" =
    Information }
    )

csg_ip requires csg_nic
ntfr requires csg_nic
```

Registry Replication (RegRep) agent

To ensure efficient failover of an application, the registry of the node on which the application was taken offline and the registry of the node on which the application is brought online must be synchronized.

The Registry Replication agent synchronizes the registry of both the nodes by monitoring the designated registry keys. When an application changes the registry keys on the active cluster node, the agent logs these changes to the shared disk. When failover occurs, the agent updates the registry of the system on which the application is brought online using the logs on the shared disk. The RegRep agent provides you the option to:

- exclude certain subkeys from being replicated.
- mark keys for replication or exclusion, even though they do not exist when the agent is configured.

- mark keys for replication or exclusion dynamically, even when the resource is online.

When the Registry Replication agent is started, it forks an independent process to monitor registry changes. Even if the VCS engine, HAD, dies, the process continues logging registry changes to the shared disk.

If you wish to replicate the registry of an application running on a node outside VCS, run the utility VCSRegUtil on that particular node.

See the *Cluster Server Administrator's Guide* for more information on VCSRegUtil utility.

Note: The Registry Replication agent runs in the context of the local system user while the `HKEY_CURRENT_USER (HKCU)` hive has values in the context of the current user. As the local system user account does not map to a current user account, the Registry Replication agent does not replicate the keys under the HKCU hive.

The RegRep agent supports intelligent resource monitoring and uses Intelligent Monitoring Framework (IMF) for resource state change notifications.

For more information about IMF, refer to the *Cluster Server Administrator's Guide*.

Dependency

The RegRep resource depends on the MountV resource.

Agent functions

online	Starts logging changes to specified registry keys to the shared disk.
offline	Stops logging changes to specified registry keys to the shared disk.
monitor	Verifies that the process that logs changes to registry keys is running.

State definitions

`ONLINE`	Indicates the agent is logging changes to specified registry keys to the shared disk.
`OFFLINE`	Indicates the agent is not logging changes to specified registry keys to the shared disk.
`UNKNOWN`	Indicates the agent could not determine the status of the resource.

Note: The Registry Replication agent is for failover service groups only. Do not use the agent for parallel applications or service groups.

Resource type definition

```
type RegRep (
    static i18nstr ArgList[] = {"MountResName:MountPath",
    ReplicationDirectory, Keys, ExcludeKeys, ListOpenKeys,
    ForceRestore, RestoreLocally, DebugMonitor }
    static int IMF{} = { Mode=2, MonitorFreq=5, RegisterRetryLimit=3 }
    str MountResName
    i18nstr ReplicationDirectory = "\\REGREP\\DEFAULT"
    i18nstr Keys{}
    i18nstr ExcludeKeys[]
    boolean ListOpenKeys = 1
    boolean ForceRestore = 1
    boolean RestoreLocally = 0
    int DebugMonitor = 1
)
```

Attribute definitions

Table 6-3 RegRep agent required attributes

Required Attributes	Description
Keys	The list of registry keys to be monitored. From the 'name-value' pair of a registry key, you must provide the name of the registry keys to be synchronized and not the value for that key. When defining the keys, you must use the abbreviations listed in See "Configuring registry keys" on page 166. The agent also enables you to replicate a registry key without replicating the subkeys.See "Ignoring subkeys" on page 166. for instructions. This attribute can take localized values. **Warning:** Do not configure more than 63 keys for a single RegRep resource otherwise the resource will go in an `UNKNOWN` state. Type and Dimension: string-association

Table 6-3 RegRep agent required attributes *(continued)*

Required Attributes	Description
MountResName	The name of the MountV resource on which the Registry Replication resource depends. The MountV resource specifies the mount drive on the shared disk where the log file is created. In case of a NetApp storage environment, this will be the name of the NetAppSnapDrive resource. Type and Dimension: string-scalar
ReplicationDirectory	The directory on the shared disk in which the registry changes are logged. The agent creates a directory with the specified name if one does not exist. Default is "\\REGREP\\DEFAULT". **Note:** This attribute can take localized values. Type and Dimension: string-scalar

Table 6-4 RegRep agent optional attributes

Optional Attributes	Description
DebugMonitor	Defines whether debug logging is to be enabled for the executable (RegRepMonitor.exe) that monitors the registry keys. The value 0 indicates that the logging is enabled. The value 1 indicates that it is not. Default is 1. Type and Dimension: integer-scalar
ExcludeKeys	A list of the subkeys to be excluded from replication. For more information on how the agent excludes keys, see Excluding keys. When defining the keys, you must use the abbreviations listed in Configuring registry keys. **Note:** This attribute can take localized values. Type and Dimension: string-vector

Table 6-4 RegRep agent optional attributes *(continued)*

Optional Attributes	Description
ForceRestore	Defines the agent behavior when the registry update fails on a node where the resource is being brought online. A registry update may fail if any of the keys being updated are open. If set to 1 and the registry update fails, the agent tries to forcibly restore the keys by: ■ backing up the configured registry keys of the node being brought online. ■ deleting the configured registry keys from the node being brought online. ■ updating the configured registry keys with the changed values. If the update fails again, the agent reapplies the original values of the registry keys from the backup. If set to 0, the agent does not try to forcibly update the registry when an update fails. Default is 0. Type and Dimension: boolean-scalar
ListOpenKeys	Defines whether the agent lists a set of registry keys, including keys having open handles, when a registry update fails. The list is logged to the agent log. The value 1 indicates the agent lists the registry key set. The value 0 indicates the agent does not. Default is 1. Type and Dimension: boolean-scalar
RestoreLocally	Defines whether the agent restores the keys on the node if it was taken offline from the same node. For example, if the agent was taken offline from SystemA and brought online again on SystemA, this attribute determines whether the registry keys would be restored on SystemA. The value 1 indicates the agent restores the keys on the system.The value 0 indicates it does not. Default is 0. Type and Dimension: boolean-scalar

Excluding keys

This section describes the algorithm the Registry Replication agent uses while excluding keys. For example, assume a registry key `KEY_X` has a subkey of `KEY_Y`, which has a subkey `KEY_Z`. This key would appear as `KEY_X\KEY_Y\KEY_Z` in the Registry Editor.

The following table describes various scenarios of keys marked for replication and for exclusion. The Result column describes the agent behavior in these scenarios.

Table 6-5 RegRep agent exclude keys and behavior

Keys for Replication	Exclude Keys	Result
KEY_X	KEY_Y\KEY_Z	KEY_Y is excluded. So is KEY_Z.
KEY_X	KEY_Y	KEY_Y is excluded. So is KEY_Z.
KEY_X	KEY_X	KEY_X is *not* excluded and an error message is logged.
KEY_X\KEY_Y	KEY_X	KEY_X is *not* excluded and an error message is logged.

Configuring registry keys

To configure a registry key to be replicated or excluded, use the abbreviation corresponding to the registry hive, as listed in the following table.

Table 6-6 RegRep agent registry hive

Registry Hive	Abbreviation
HKEY_LOCAL_MACHINE	HKLM
HKEY_USERS	HKU
HKEY_CURRENT_CONFIG	HKCC
HKEY_CLASSES_ROOT	HKCR

Ignoring subkeys

Use the IgnoreSubKeys option for the Keys attribute to prevent the RegistryReplication agent from replicating the subkeys.

The following table describes possible combination of values for the Keys attribute. The Result column describes the agent behavior in these scenarios:

Table 6-7 RegRep agent IgnoreSubKeys and behavior

Value Specified for "Keys" Attribute	Result
"HKLM\SOFTWARE\VERITAS\VCS"	Replicates the subkeys

Table 6-7 RegRep agent IgnoreSubKeys and behavior *(continued)*

Value Specified for "Keys" Attribute	Result
"HKLM\SOFTWARE\VERITAS\VCS"=IgnoreSubKeys	Does not replicate the subkeys
"HKLM\SOFTWARE\VERITAS\VCS"=IgnoreSubKeys:Yes	Does not replicate the subkeys
"HKLM\SOFTWARE\VERITAS\VCS"=IgnoreSubKeys:No	Replicates the subkeys
"HKLM\SOFTWARE\VERITAS\VCS"=<any other value>	Replicates the subkeys

Additional considerations for using IgnoreSubKeys

Veritas recommends not to set the "IgnoreSubKeys" value when the RegRep resource is online. Even if the value is set with the resource online, the changes will be applicable after the next online routine.

Sample configuration

In the following sample configuration, the Registry Replication agent is configured to replicate the print spooler registry key.

```
VMDg VMDg-1 (
    DiskGroupName = { VCSDG }
    )

IP IP-10-141-15-49 (
    Address = "10.141.15.49"
    SubNetMask = "255.255.248.0"
    MACAddress @W2K-VCS101 = "00-02-B3-1A-13-1E"
    MACAddress @W2K-VCS102 = "00-02-B3-1A-17-48"
    )

Lanman Lanman-TESTSQL (
    VirtualName = TESTSQL
    IPResName = IP-10-141-15-49
    )

MountV MountV-HDrive (
    MountPath = "H:"
    VMDGResName = VMDg-1
    VolumeName = Volume1
```

```
    )

MountV MountV-RDrive (
    MountPath = "R:"
    VMDGResName = VMDg-1
    VolumeName = Volume2
    )

RegRep SQL2008-RegRep (
    MountResName = MountV-RDrive
    Keys = {
    "HKLM\\Software\\Microsoft\\Microsoft SQL Server\\MSSQL.1
     \\MSSQLServer"="SaveRestoreFile:SQL2008-RegRep_MSSQLServer.reg",
    "HKLM\\Software\\Microsoft\\Microsoft SQL Server\\MSSQL.1
     \\Replication"="SaveRestoreFile:SQL2008-RegRep_Replication.reg",
    "HKLM\\Software\\Microsoft\\Microsoft SQL Server\\MSSQL.1
     \\SQLServerAgent"="SaveRestoreFile:SQL2008-RegRep_SQLServerAgent.reg",
    "HKLM\\Software\\Microsoft\\Microsoft SQL Server\\MSSQL.1
     \\PROVIDERS"="SaveRestoreFile:SQL2008-RegRep_PROVIDERS.reg",
    "HKLM\\Software\\Microsoft\\Microsoft SQL Server\\MSSQL.1
     \\SQLServerSCP"="SaveRestoreFile:SQL2008-RegRep_SQLServerSCP.reg"
     }
    )

NIC NIC-test (
    MACAddress @W2K-VCS101 = "00-02-B3-1A-13-1E"
    MACAddress @W2K-VCS102 = "00-02-B3-1A-17-48"
    )

SQLServer2008 SQLServer2008-Default (
    LanmanResName = Lanman-TESTSQL
    )

IP-10-141-15-49 requires NIC-test
Lanman-TESTSQL requires IP-10-141-15-49
MountV-HDrive requires VMDg-1
MountV-RDrive requires VMDg-1
SQLServer2008-Default requires MountV-HDrive
SQLServer2008-Default requires Lanman-TESTSQL
SQLServer2008-Default requires SQL2008-RegRep
SQL2008-RegRep requires MountV-RDrive
```

Proxy agent

The Proxy agent monitors and mirrors the state of a resource on a local or remote system. The agent can be used to reduce monitoring overheads in configurations where multiple resources point at the same physical device. For example, if multiple service groups use the same NIC, configure one service group to monitor the NIC and have Proxy resources in the other service groups to mirror the state of the NIC resource. The agent can also determine the status of an OnOff resource in a different service group.

Veritas recommends the use of Proxy agent to monitor persistent resources such as NIC, ServiceMonitor, ElifNone, and FileNone.

Agent functions

online	Not applicable.
offline	Not applicable.
monitor	Determines the status of the target resource.

State definitions

`ONLINE`	Indicates the target resource is online.
`FAULTED`	Indicates the target resource is offline or faulted.

Resource type definition

```
type Proxy (
    static int OfflineMonitorInterval = 60
    static i18nstr ArgList[] = { TargetResName, TargetSysName,
    "TargetResName:Probed", "TargetResName:State"}
    static str Operations = None
    str TargetResName
    str TargetSysName
)
```

Attribute definitions

Table 6-8 Proxy agent required attributes

Required Attribute	Description
TargetResName	The name of the target resource whose status is monitored and mirrored by the Proxy resource. Type and Dimension: string-scalar

Table 6-9 Proxy agent optional attributes

Optional Attribute	Description
TargetSysName	The name of the system associated with the target resource. If this attribute is not specified, the Proxy resource assumes the system is local. Type and Dimension: string-scalar

Sample configurations

```
group  NicGrp (
        SystemList = { SYSTEM1, SYSTEM2 }
        Parallel = 1
)

Phantom my_phantom (
)

NIC  NicGrp_public (
    MACAddress @SYSTEM1 = "00-50-04-24-F9-60"
    MACAddress @SYSTEM2 = "00-50-04-24-E3-30"
    UseConnectionStatus    = True
)

group Grp1 (
    SystemList = { SYSTEM1,SYSTEM2 }
    AutoStartList = { SYSTEM1 }
)
NotifierMngr ntfr(
    SnmpConsoles = { your_console = Information }
    SmtpServer   = "smtp.your_company.com"
    SmtpRecipients = { "vcsadmin@your_company.com" =
```

```
    Information }
)

Proxy nicproxy (
    TargetResName    =    "NicGrp_public"
)

ntfr requires nicproxy
```

Phantom agent

The Phantom agent enables VCS to determine the status of parallel service groups that do not include OnOff resources (resources that VCS starts and stops as required.) The Phantom agent provides a "dummy" resource that can be brought online and taken offline. Without such a dummy resource, VCS cannot assess the status of service groups containing only None (Persistent) and OnOnly resources because the state of these resources is not considered in the process of determining whether a group is online.

Agent functions

online	Not applicable.
offline	Not applicable.
monitor	Determines status based on the status of its service group.

State definitions

`ONLINE`	Indicates the service group is online.
`OFFLINE`	Indicates the service group is offline.

Attribute definitions

Not applicable.

Sample configuration

```
FileNone FILE_1 (
    PathName = "C:\\Temp\\file1"
    )
```

```
FileOnOnly FILE_2 (
    PathName = "C:\\Temp\\file2"
    )

ElifNone FILE_3 (
    PathName = "C:\\Temp\\file3"
)

Phantom PHANTOM_1 (
)
```

FileNone agent

The FileNone agent monitors a file. The monitor routine returns `ONLINE` if the specified file exists.

Agent functions

online	Not applicable.
offline	Not applicable.
monitor	Verifies that the specified file exists.

State definitions

`ONLINE`	Indicates the specified file exists.
`FAULTED`	Indicates the specified file does not exist.
`UNKNOWN`	Indicates that the value of the PathName attribute does not contain a file name.

Resource type definition

```
type FileNone (
    static i18nstr ArgList[] = { PathName }
    i18nstr PathName
)
```

Attribute definitions

Table 6-10 FileNone agent required attributes

Required Attributes	Description
PathName	The complete path of the file to be monitored. This attribute can take localized values. Type and Dimension: string-scalar

Sample configuration

```
FileNone tmp_file01_txt (
    PathName = "C:\\tmp\\file01.txt"
)
```

ElifNone agent

The ElifNone agent monitors a file. The monitor routine returns `ONLINE` if the specified file does not exist.

Agent functions

online	Not applicable.
offline	Not applicable.
monitor	Verifies that the specified file exists.

State definitions

`ONLINE`	Indicates the specified file does not exist.
`FAULTED`	Indicates the specified file exists.
`UNKNOWN`	Indicates that the value of the PathName attribute does not contain a file name.

Resource type definition

```
ElifNone (
    static i18nstr ArgList[] = { PathName }
```

```
    i18nstr PathName
)
```

Attribute definitions

Table 6-11 ElifNone agent required attributes

Required Attributes	Description
PathName	The complete path of the file to be monitored. This attribute can take localized values. Type and Dimension: string-scalar

Sample configuration

```
ElifNone tmp_file01_txt (
    PathName = "C:\\tmp\\file01.txt"
)
```

FileOnOff agent

The FileOnOff agent creates, removes, and monitors files.

Agent functions

online	Creates an empty file with the specified name if the file does not already exist.
offline	Removes the specified file.
monitor	Verifies that the specified file exists.

State definitions

ONLINE	Indicates the specified file exists.
OFFLINE	Indicates the specified file does not exist.
UNKNOWN	Indicates that the value of the PathName attribute does not contain a file name.

Resource type definition

```
type FileOnOff (
    static i18nstr ArgList[] = { PathName }
    i18nstr PathName
)
```

Attribute definitions

Table 6-12 FileOnOff agent required attributes

Required Attributes	Description
PathName	The complete path of the file to be monitored. This attribute can take localized values. Type and Dimension: string-scalar

Sample configuration

```
FileOnOff tmp_file01_txt (
    PathName = "C:\\tmp\\file01.txt"
)
```

FileOnOnly agent

The FileOnOnly agent creates and monitors a file.

Agent functions

online	Creates the specified file.
offline	Not applicable.
monitor	Verifies that the specified file exists.

State definitions

ONLINE	Indicates the specified file exists.
FAULTED	Indicates the specified file does not exist.

`UNKNOWN`	Indicates that the value of the PathName attribute does not contain a file name.

Resource type definition

```
type FileOnOnly (
    static i18nstr ArgList[] = { PathName }
    i18nstr PathName
)
```

Attribute definitions

Table 6-13 FileOnOnly agent required attributes

Required Attributes	Description
PathName	The complete path of the file to be monitored. This attribute can take localized values. Type and Dimension: string-scalar

Sample configuration

```
FileOnOnly tmp_file01_txt (
    PathName = "C:\\tmp\\file01.txt"
)
```

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 are:

- 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.

Veritas InfoScale products support 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 *Cluster Server Administrator's Guide*.

Dependency

As a best practice, establish a RemoteGroup resource dependency on a NIC resource. Veritas recommends that the RemoteGroup resource not be by itself in a service group.

Agent functions

Online	Depending on the ControlMode setting, this function brings the remote service group online.
Offline	Depending on the ControlMode setting, this function takes the remote service group offline.
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.
Clean	Depending on the ControlMode setting, Clean takes the remote service group offline whenever the RemoteGroup resource faults.

State definitions

ONLINE	Indicates that the remote service group is either in an ONLINE or PARTIAL state. If the ReturnIntOffline attribute is not set to RemotePartial, then the remote service group is either in an ONLINE or PARTIAL state.
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.
FAULTED	Indicates that the remote service group 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.

Resource type definition

```
type RemoteGroup (
    static int OnlineRetryLimit = 2
    static int ToleranceLimit = 1
    static i18nstr ArgList[] = { IpAddress, Port, Username,
    Password, GroupName, VCSSysName, ControlMode, OfflineWaitTime,
    DomainType, BrokerIp }
    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

)
```

Attribute definitions

Table 6-14 RemoteGroup agent required attributes

Required Attribute	Description
IpAddress	The IP address or DNS name of a node in the remote cluster, with which the agent communicates to get the status of the remote service group. The IP address can be either physical or virtual. If the IP address is a virtual IP address, make sure that this virtual IP address is not configured as an IP resource in the remote service group that is to be monitored. Example: "www.example.com" or "11.183.12.214" Type and Dimension: string-scalar
Port	The port on the remote node with which the agent communicates. This is an optional attribute, unless the remote cluster listens on another port. The default value is 14141. Type and Dimension: integer-scalar
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 in the Java Console to enter your username data, you need to enter your username and the domain name in separate fields. For a cluster that has the VCS Authentication Service, you do not need to enter the domain name. Type and Dimension: string-scalar
Password	This is the password that corresponds to the user that you specify in the Username attribute. You must encrypt the password using the VCS Encrypt utility. For more information about the VCS Encrypt utility, see the *Cluster Server Administrator's Guide*. Do not use the vcsencrypt utility when entering passwords from a configuration wizard or from the Cluster Management Console or the Cluster Manager (Java Console). Type and Dimension: string-scalar

Table 6-14 RemoteGroup agent required attributes *(continued)*

Required Attribute	Description
GroupName	The name of the service group on the remote cluster that you want the RemoteGroup agent to monitor or manage. Example: "DBGrp" Type and Dimension: string-scalar
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. Example: "vcssys1" or "ANY" Type and Dimension: string-scalar

Table 6-14 RemoteGroup agent required attributes *(continued)*

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 comes online. ■ MonitorOnly The RemoteGroup resource only monitors the state of the remote service group. The RemoteGroup resource cannot bring the remote service group online or take it offline. Make sure that you bring the remote service group online before you bring the RemoteGroup resource online. ■ 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 comes online. Type and Dimension: string-scalar

Table 6-15 RemoteGroup agent optional attributes

Optional Attribute	Description
OfflineWaitTime	The maximum expected time in seconds that the remote service group may take to go 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. The default value is 0. Type and Dimension: integer-scalar

Table 6-15 RemoteGroup agent optional attributes *(continued)*

Optional Attribute	Description
DomainType	Enter the domain type information for the specified user. This is for a secure remote cluster only. For users who have the domain type unixpwd, you do not have to set this attribute. Example: "nis", "nisplus" Type and Dimension: string-scalar
BrokerIp	For a secure remote cluster only, if the user needs the RemoteGroup agent to communicate to a specific authentication broker, then set this attribute. Enter the information for the specific authentication broker in the format "IP". Example: "128.11.295.51" Type and Dimension: string-scalar
OnlineRetryLimit OnlineWaitLimit ToleranceLimit MonitorInterval AutoFailover	In case of remote service groups that take a longer time to come online, Veritas recommends that you modify the default OnlineWaitLimit and OnlineRetryLimit attributes. If you expect the RemoteGroup agent to tolerate sudden offlines of the remote service group, then modify the ToleranceLimit attribute. See the *Cluster Server Administrator's Guide* for more information about these attributes. Type and Dimension: integer-scalar

Table 6-15 RemoteGroup agent optional attributes *(continued)*

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. Default: "" Type and Dimension: string-vector
OfflineMonitoringN ode	Defines the cluster node that performs the offline monitoring of the remote service group. This is an internal attribute. Do not modify.

About the ControlMode attribute

In the ControlMode attribute, you can use these values, depending on your needs: OnOff, MonitorOnly, and OnlineOnly.

OnOff

Select the OnOff value of this attribute when you want the RemoteGroup resource to manage the remote service group completely.

The username provided in the Username attribute should have administrator privileges for the service group on the remote cluster.

In case of one-to-one mapping, set the value of the AutoFailover attribute of the remote service group to 0. This avoids unnecessary onlining or offlining of the remote service group.

MonitorOnly

Select the MonitorOnly value of this attribute when you want to monitor the state of the remote service group. When you choose the MonitorOnly attribute, the RemoteGroup agent does not have control over the remote service group and cannot bring it online nor take it offline.

The remote service group should be in an `ONLINE` state before you bring the RemoteGroup resource online.

Veritas recommends that you set the AutoFailover attribute of the remote service group to 1.

OnlineOnly

Select the OnlineOnly value of this attribute when the remote service group takes a long time to come online or to go offline. When you use OnlineOnly for the ControlMode attribute, a switch or fail over of the local service group with VCSSysName set to ANY does not cause the remote service group to be taken offline and brought online.

The username provided in the Username attribute needs to have administrator privileges for the remote service group.

The Offline of the RemoteGroup resource does not offline the remote service group.

If you are choosing one-to-one mapping between the local nodes and remote nodes, then the value of the AutoFailover attribute of the remote service group must be 0.

Note: When you set the value of ControlMode to OnlineOnly or to MonitorOnly, the recommended value of the VCSSysName attribute of the RemoteGroup resource is ANY. If you want one-to-one mapping between the local nodes and the remote nodes, then a switch or fail over of local service group is impossible. It is important to note that in both these configurations the RemoteGroup agent does not take the remote service group offline.

Sample scenario

In this example, a local VCS cluster (front-end cluster) provides high availability for Web-services and a remote VCS cluster (back-end cluster) provides high availability for the database required by the Web-services. The cluster setup is as follows:

- The local cluster is a two-node cluster, where local1 and local2 are the names of the two local nodes.
- The remote cluster is a two-node cluster, where remote1 and remote2 are the names of the two remote nodes.

- LocGrp is the local service group that contains the WebRes resource (for example: Apache, etc.) configured for the Web-server that runs in the local cluster.
- RemGrp is the remote service group that contains the DBRes resource (for example: MySQL or Oracle, etc.) configured for the database in the remote cluster.

The database resource (DBRes) needs to come online before the Web server resource (WebRes) comes online. You can use the RemoteGroup agent to establish this inter-cluster application dependency.

To configure the remotegroup agent for this setup

1. Create a RemoteGroup resource (RGRes) in the local service group (LocGrp) for monitoring or managing the DBRes resource in the remote cluster.
2. Link the resources such that the WebRes resource depends on the RGRes resource.
3. Set the attributes for the RemoteGroup resource (RGRes) as follows:
   - **IpAddress**—Set to the IP address or DNS name of a node in the remote cluster. For this example, the value is remote1. You can also set this value as remote2 (second node in the remote cluster).
     The RemoteGroup agent keeps a list of remote nodes (remote1, remote2) on the local node. The IP address that you specify should be reachable the first time the RemoteGroup agent tries to communicate to the remote cluster. In case the specified IP address is not reachable after the first communication, the RemoteGroup agent talks to the second remote node to retrieve the state of RemGrp.
   - **GroupName**—Set the value to RemGrp (the remote service group).
   - **VCSSysName**—Set appropriate values so as to establish a one-to-one mapping between the systems in the local and remote clusters. RGRes will completely manage the remote service group (RemGrp) that contains the DBRes resource. Make this a local attribute for the RGRes.
   - **ControlMode**—Set the value to OnOff.
   - **Username**—Set the value to user_admin. The user needs to have administrative privileges for RemGrp.
   - **Password**—Password for user_admin. You must encrypt the password using the VCS Encrypt command. See the *Cluster Server Administrator's Guide* for more information on the VCS Encrypt utility.
   - **VCSSysName@local1**—Set this value to remote1.
   - **VCSSysName@local2**—Set this value to remote2.

You have to set the value of BrokerIp for a remote cluster that has the VCS Authentication Service enabled.

4 Set the value of the AutoFailover attribute of the RemGrp to 0. This avoids unnecessary online or offline of the remote service group in case of a fail over. If the value of the VCSSysName attribute is ANY, then the value of the AutoFailover attribute of RemGrp should be 1.

Service group behavior

Consider the following actions to better understand the service group behavior in this configuration.

- **Bringing the LocGrp local service group online**
  The Online functions of all the resources in LocGrp (local service group) are called starting from the lower-most resource on local1. In this configuration, WebRes depends on RGRes. The RGRes Online function is called before the WebRes Online function. In the Online function, the RemoteGroup agent communicates to the remote cluster and authenticates the specified user. The RemoteGroup agent brings RemGrp online on remote1. RemGrp is in an ONLINE state on remote1 and RGRes is in an ONLINE state on local1. The Online function of WebRes is then called. WebRes is now in an ONLINE state.
  Thus, you have established an application-level dependency across two different VCS clusters. The WebRes does not go online before the RGRes goes online. RGRes does not go online before the RemGrp goes online.
- **An unexpected offline of the RemGrp remote service group**
  LocGrp is online on the local1. WebRes and RGRes are both online on local1. RemGrp is online on remote1. If RemGrp faults on remote1, RGRes detects that RemGrp has gone OFFLINE or has FAULTED on remote1 and goes into a FAULTED state. All the resources in LocGrp are taken offline on local1. LocGrp now fails over to local2. All the resources including WebRes and RGRes go into an ONLINE state on local2. As part of the fail over of LocGrp on local2, RemGrp goes online on remote2. LocGrp shows ONLINE on node local2 and OFFLINE OR FAULTED on local1.
- **Taking the LocGrp service group offline on local1**
  LocGrp is online on local1 and RemGrp is online on remote1. If you take LocGrp offline, all the resources dependent on RGRes are taken offline. First, the WebRes resource goes offline and then the offline for RGRes is called. As the ControlMode is OnOff, the RemoteGroup agent communicates to the remote cluster and issues an offline command for RemGrp. Once RemGrp has gone offline, RGRes goes into OFFLINE state. Thus, the Web-server is taken offline before the database goes offline.

Chapter 7

VCS hardware replication agent for NetApp

This chapter includes the following topics:

- About hardware replication agents for NetApp
- NetApp Filer agent
- NetApp SnapDrive agent
- NetApp SnapMirror agent

About hardware replication agents for NetApp

The VCS hardware replication agent for Network Appliance provides failover support and recovery, in environments employing Network Appliance filers for storage and SnapMirror for replication.

The agent monitors and manages the state of replicated filer devices and ensures that at a time only one system has safe and exclusive access to the configured devices.

The agent can be used in local clusters, single VCS replicated data clusters, and multi-cluster environments set up using the VCS Global Cluster Option.

The package contains three agents; each agent is described in subsequent sections:

- NetAppFiler agent—Monitors the state of the filer.
- NetAppSnapDrive agent—Connects and disconnects virtual disks (LUNs) using the iSCSI or FC protocol.
- NetAppSnapMirror agent—Determines the role of the filer volumes with respect to replication and promotes a read-only snapmirrored volume to a read-write source volume during a wide-area failover.

Note: The NetApp agents are available with Cluster Server for Windows and require a separate HA/DR license.

NetApp Filer agent

The NetApp Filer agent monitors the state of the filer device. The NetApp Filer agent is configured as a resource of type NetAppFiler. NetAppFiler resources are persistent, meaning that they are not brought online or taken offline.

Agent function

Monitor — Performs the following tasks:

- Verifies the state of the filer attached to the host by sending an ICMP ping command to the filer. If the filer does not respond, the agent reports the state of the filer as faulted.
- Opens a filer connection and checks if ONTAPI version is supported by the filer. If the connection fails or the ONTAPI version is not supported, the agent reports the state as offline.

Resource type definition

```
type NetAppFiler (
        static int MonitorInterval = 30
        static i18nstr ArgList[] = { FilerName, StorageIP }
        static str Operations = None
        str FilerName
        str StorageIP
)
```

Attribute definitions

Table 7-1 NetApp Filer agent attributes

Required Attribute	Description
FilerName	DNS-resolvable name or IP address of the locally attached filer. Type and Dimension: string-scalar
StorageIP	The private storage IP address of the filer. Type and Dimension: string-scalar

Sample configuration

```
NetAppFiler A2-NetAppFiler (
    FilerName = winvm-netapp02
    StorageIP = "10.121.57.8"
    )
```

NetApp SnapDrive agent

The NetApp SnapDrive agent monitors, connects, and disconnects filer volumes. The agent can be configured to use the iSCSI or the FC protocol.

Note: This agent supports 7-mode configurations only. C-mode configurations are not supported.

NetApp SnapDrive agent is configured as a resource of type NetAppSnapDrive.

Dependency

The NetApp SnapDrive resource depends on the NetApp SnapMirror and the NetApp Filer resources for operation.

If the NetApp SnapMirror resource is not configured, the NetApp SnapDrive resource depends on the NetApp Filer resource.

Agent functions

Online	Connects a virtual disk (LUN) using an iSCSI or an FC initiator. The agent presents the LUN as a locally-attached drive to the host. The agent also removes LUN-host mappings made before the online operation.
Offline	Disconnects the virtual disk (LUN) from the host.
Monitor	Verifies that the specified virtual disk (LUN) is connected to the host.
Open	Verifies that there is connectivity to the filer. It also checks that the VCS Helper service is running with the same privileges as the SnapDrive service.
Clean	Attempts to forcibly disconnect a virtual disk (LUN).

Resource type definition

```
type NetAppSnapDrive (
    static int MonitorInterval = 30
    static int NumThreads = 1
    static i18nstr ArgList[] = { FilerResName,
    "FilerResName:FilerName", "FilerResName:StorageIP",
    VolumeName, ShareName, LUN, MountPath, Initiator,
    InitiatorMonitorInterval }
    str FilerResName
    str VolumeName
    str ShareName
    str LUN
    str MountPath
    str Initiator[]
    int InitiatorMonitorInterval = 30
)
```

Attribute definitions

Table 7-2 NetApp SnapDrive agent attributes

Required Attribute	Description
FilerResName	Name of the VCS NetAppFiler-type resource in the service group. Type and Dimension: string-scalar
VolumeName	Name of the volume containing the virtual disk. Define the volume name in the same case as on the filer. Type and Dimension: string-scalar
ShareName	Name of the CIFS share containing the virtual disk. Type and Dimension: string-scalar
LUN	Name of the LUN (virtual disk) on the filer that is presented to the host for mounting. Define the LUN name in the same case as on the filer. Type and Dimension: string-scalar
MountPath	Drive letter to be assigned to the virtual disk. Type and Dimension: string-scalar

Table 7-2 NetApp SnapDrive agent attributes *(continued)*

Required Attribute	Description
Initiator	Name of iSCSI or FC initiator the host uses to connect virtual disks. You can retrieve this value from the Disk Management console. Type and Dimension: string-vector

Sample configuration

```
NetAppSnapDrive A2-NetAppSnapDrive (
    FilerResName = A2-NetAppFiler
    VolumeName = vol8
    ShareName = data08
    LUN = "d85.lun"
    MountPath = T
    Initiator @VCSW2K288 = { "21:00:00:e0:8b:1c:cc:45",
         "21:00:00:e0:8b:90:74:1a" }
    Initiator @VCSW2K287 = { "21:00:00:e0:8b:0e:3a:6a" }
    )
```

NetApp SnapMirror agent

The NetApp SnapMirror agent monitors the replication state of filer devices. When a failover occurs, the agent reverses the direction of replication.

NetApp SnapMirror agent is configured as a resource of type NetAppSnapMirror.

Dependency

The NetApp SnapMirror resource depends on the NetApp Filer resource for operation.

Agent functions

Online	If the state of the local filer device is `SOURCE`, the agent creates a lock file to indicate that the resource can come online. This effectively makes the devices writable for the application. If the state of the local filer is `SNAPMIRRORED`, the agent attempts to reverse the direction of replication by changing the state of the local filer to `SOURCE` and that of the original source to `SNAPMIRRORED`. If the original source filer is down, the agent performs a mirror break-off to enable local write access, if the filer is not already broken off. If the original source returns to life, you must resynchronize the data manually.
Offline	Removes the lock file. The agent does not perform any filer operations because an offline function does not necessarily indicate an intention to give up the devices.
Monitor	Verifies the lock file exists. If the lock file exists, the monitor function reports the status of the resource as online. If the lock file does not exist, the monitor function reports the status of the resource as offline.
Open	Removes the lock file, thereby preventing potential concurrency violation if the group fails over to another node. **Note:** The agent does not remove the lock file if the agent was started after an `hastop -force` command.
Clean	Removes the lock file. No filer operations are performed since offlining this resource is no indication of a pending role swap.

Action function

The Action function enables you to perform predefined actions on a resource.

To perform an action on a resource, type the following command:

```
hares -action <SnapMirror_resname> <token> [-actionargs
<arg1> ...] [-sys <system>] [-clus <cluster> ]
```

The following table lists the action supported by the NetAppSnapMirror agent:

Table 7-3 Actions supported by NetAppSnapMirror agent

Token for Action	Description
fbsync	Resynchronizes an original source volume with a broken-off volume. After synchronization, the original source volume becomes the target volume. The broken-off volume was initially the target volume, but was broken off as a result of a take over.

To synchronize volumes, type the following at the command prompt:

```
hares -action SnapMirror_resname fbsync -sys node_name
```

Where, *SnapMirror_resname* represents the name of the SnapMirror resource and *node_name* represents the node on which the service group is online.

Run the action for each SnapMirror resource.

You can also add custom actions for the agents. Refer to the *Cluster Server Agent Developer's Guide* for more information.

Resource type definition

```
type NetAppSnapMirror (
    static keylist SupportedActions = { fbsync }
    static int MonitorInterval = 300
    static int NumThreads = 1
    static i18nstr ArgList[] = { FilerResName,
    "FilerResName:FilerName",
    "FilerResName:StorageIP", VolumeName, SnapMirrorArguments,
    SnapMirrorSchedule, AppResName, VisibilityFrequency,
    SyncMode }
    str FilerResName
    str VolumeName
    str SnapMirrorArguments
    str SnapMirrorSchedule
    str AppResName
    int VisibilityFrequency = 180
    str SyncMode = async
)
```

Attribute definitions

Table 7-4 NetApp SnapMirror agent attributes

Required Attribute	Description
FilerResName	Name of the VCS NetAppFiler-type resource in the group. Type and Dimension: string-scalar
VolumeName	Name of the volume containing the virtual disk. Define the volume name in the same case as on the filer. Type and Dimension: string-scalar
SnapMirrorArguments	Specifies the SnapMirror arguments such as maximum transfer speed and restart mode. The format for this attribute is: **<RestartMode> MaxSpeed** MaxSpeed is an optional parameter. The RestartMode parameter can have the following values: ■ Always ■ Never ■ Default Setting the RestartMode parameter value to default indicates schedule priority. Default value of RestartMode parameter is ‘default’ Example: Always MaxSpeed Type and Dimension: string-scalar
SnapMirrorSchedule	Specifies the schedule the destination uses for updating data. Do not assign a value for this attribute if you use SnapManager. The schedule is in the following format: **minute hour dayofmonth dayofweek** Each field is separated by a space. Refer to the NetApp documentation for more details on the rules for each of these schedule fields. By default, this attribute does not have any value. Type and Dimension: string-scalar

Table 7-4 NetApp SnapMirror agent attributes *(continued)*

Required Attribute	Description
AppResName	Name of the resource configured to monitor the application being made highly available. Type and Dimension: string-scalar
SyncMode	Specifies the mode of replication for the mirror. This attribute can have the following values: ■ async: Indicates that the mirror should be configured in the asynchronous mode. ■ semi-sync: Indicates that the mirror should be configured in the semi-synchronous mode. ■ sync: Indicates that the mirror should be configured in the synchronous mode. The default is async (asynchronous) mode. Type and Dimension: string-scalar
VisibilityFrequency	Specifies how often the source snapshot will be visible on the destination mirror. It controls the value of visibility_interval in the snapmirror.conf file. The default value is 180 seconds. This attribute is applicable only if the mirror is configured in synchronous or semi-synchronous mode. Type and Dimension: string-scalar

Sample configuration

```
NetAppSnapMirror sqlg-NetAppSnapMirror (
    FilerResName = sqlg-NetAppFiler
    VolumeName = vol3
    )
```

Chapter 8

Troubleshooting bundled agents

This chapter includes the following topics:

- About troubleshooting bundled agents
- VCS logging
- Troubleshooting storage agents
- Troubleshooting network agents
- Troubleshooting FileShare agents
- Troubleshooting services and application agents
- Troubleshooting infrastructure and support agents
- Troubleshooting NetApp agents

About troubleshooting bundled agents

This chapter describes how to troubleshoot common problems in bundled agents. The chapter lists bundled agent error messages (shown in bold text), and describes the problems associated with each agent. Recommended action is included, where applicable.

This chapter lists most of the error messages associated with the VCS bundled agents. For error messages other than the ones described here, contact Veritas Technical Support.

VCS logging

VCS generates two error message logs: the engine log and the agent log. Log file names are appended by letters. Letter A indicates the first log file, B the second, C the third, and so on.

The agent log is located at %VCS_HOME%\log*agent*_A.txt. The format of agent log messages is:

Timestamp (Year/MM/DD) | Mnemonic | Severity | UMI | Agent Type | Resource Name | Function | Message Text

The agent log message components are defined as follows:

- *Timestamp*: the date and time the message was generated.
- *Mnemonic*: the string ID that represents the product (for example, VCS).
- *Severity*: levels include `CRITICAL`, `ERROR`, `WARNING`, `NOTICE`, and `INFO` (most to least severe, respectively).
- *UMI*: a unique message ID.
- *Message Text*: the actual message generated by VCS.

A typical agent log resembles:

```
2004/03/02 16:34:07 VCS WARNING V-16-10051-5008
NotifierMngr:ntfr:monitor:Expected correct SNMP and | or SMTP
options
```

Troubleshooting storage agents

Storage agents comprise the Volume Manager Diskgroup agent and MountV agent.

Note: In case of a NetApp storage environment, the Logical Unit Numbers (LUNs) on the NetApp Filer are made highly available by the NetApp Filer and NetApp SnapDrive agents instead of the VMDg and MountV agents respectively. Refer to the application-specific VCS implementation guides for more information on these agents.

Disk Reservation agent error messages

No arguments

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Insufficient arguments

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Insufficient memory

This is a VCS internal error.

Recommended Action: Close applications not is use and check memory usage. Contact Veritas Technical Support if the problem persists.

Invalid signature(s) specified. One of the invalid signature is :*signature*

Invalid disk signature specified in the configuration

Recommended Action: Verify the Signature attribute defined in the configuration file is assigned a correct value.

Signature(s) not specified

No value specified for the Signature attribute.

Recommended Action: Verify the Signature attribute defined in the configuration file is assigned a value.

Duplicate signature(s) specified

Duplicate values specified for the Signature attribute.

Recommended Action: Verify the Signature attribute defined in the configuration file is assigned a unique value.

Invalid ConfigPercentage specified

Invalid value specified for the ConfigPercentage attribute.

Recommended Action: Verify the value specified for ConfigPercentage attribute ranges between 51 and 100.

Insufficient memory to store the ConfigPercentage

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Insufficient memory to store the FailFastInterval

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Insufficient memory to store the GracePeriod

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Invalid ConfigPercentage specified. Setting it to the default value (*default_percentage*)

The value specified for the attribute ConfigPercentage is less than the minimum value, 51. The agent will set the value to 80, which is the default value.

Invalid GracePeriod specified. Setting it to the default value (*default_grace_period*)

The value specified for the attribute GracePeriod is invalid. The agent will set the value to 5, which is the default value.

VCS Disk reservation driver is not running

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

DiskRes agent has lost the majority.(Acquired = *percentage* Configured = *percentage*)

The agent could not acquire reservation to the number of disks equaling the percentage set in the attribute ConfigPercentage.

Failed to release the disk (Signature = *disk_signature*). Windows error code: *Error_code*

The agent failed to release the disk with the specified signature.

Recommended Action: See the corresponding error code for more information. Contact Veritas Technical Support if the problem persists.

Could not connect to the local node. Error : *Error_code*

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Insufficient memory to store disk ids. Error : *Error_code*

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Failed to get the disk list. Error : *Error_code*

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Failed to get the Port number and Path ID for disk *disk_signature*. Error : *Error_code*

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Failed to reset the Port *port_no* (Path ID *path*) for disk disk_signature. Error :*Error_code*

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Failed to reset the disk(s). Error : *Error_code*

The agent failed to reset the SCSI disk path. An I/O error or the driver DiskRes.Sys is not running.

Recommended Action: Verify that the driver DiskRes.Sys is located at %WINDIR%\System32\Drivers. Verify that the disk is accessible. See the corresponding error code for more information. Contact Veritas Technical Support if the problem persists.

Failed to create the disk information. Error : *Error_code*

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Failed to rescan the devices. Error : *Error_code*

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Failed to create the disk information even after rescan. Error : *Error_code*

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Failed to reserve *number_of_disks* disk(s) out of *number_of_disks* disk(s) configured.

Error: *Error_code*

The agent failed to reserve the specified number of disks out of the total configured disks.

Recommended Action: Verify the configuration definition of the agent. Verify that no other system holds reservation to the disk. Make sure that the driver DiskRes.Sys is located at %WINDIR%\System32\Drivers.

Failed to release *number_of_disks* disk(s) out of all disks configured.

Error:*Error_code*

The agent failed to release the specified number of disks out of the total configured disks.

Recommended Action: Verify the configuration definition of the agent. Make sure that the driver DiskRes.Sys is located at %WINDIR%\System32\Drivers.

Failed to release *number_of_disks* disk(s) out of *number_of_disks* disks configured.

Error: *Error_code*

The agent failed to release the specified number of disks out of the total configured disks.

Recommended Action: Verify the configuration definition of the agent. Make sure that the driver DiskRes.Sys is located at %WINDIR%\System32\Drivers.

Failed to create the disk information. Windows error code: *Error_code*

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

***Percentage* disk(s)- *number_of_disks* out of *number_of_disks* disks configured are still persistently reserved. The resource cannot failover.**

The specified number of disks are reserved on the node where the DiskRes resource is online. The resource cannot failover to another node.

Failed to get the disk count. Error : *Error_code*

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Failed to stop the driver. Error : *Error_code*

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

The agent is shutting down but there are still *no_of_disks* disk(s) under persistent reservation.

Some disks are still persistently reserved by the DiskRes resource while the agent is shutting down.

Error occurred while getting the disk reservation cache information. Error *Error_code*

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Could not collect the cache information for disk *disk_no*. Error : *Error_code*

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Failed update the cache information. Error: *Error_code*

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

DiskCache information *%s* failed for disk *disk_signature*. Error: *Error_code*

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Mount agent error messages

Unable to get the arguments

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Drive path not specified

Incorrect or no value specified for the MountPath attribute.

Recommended Action: Verify that the configuration definition for the MountPath attribute specifies the correct mount path.

Partition number not specified

Incorrect or no value specified for the PartitionNo attribute.

Recommended Action: Verify that the configuration definition for the PartitionNo attribute specifies the correct partition number.

Drive signature not specified

Incorrect or no value specified for the Signature attribute.

Recommended Action: Verify that the configuration definition for the Signature attribute specifies the correct disk signature.

Invalid disk signature (*disk_signature*) specified

Incorrect value specified for the Signature attribute.

Recommended Action: Verify that the configuration definition for the Signature attribute specifies the correct disk signature.

Invalid Child mount path specified

Incorrect value specified for the MountResName attribute.

Recommended Action: Verify that the configuration definition for the MountResName attribute specifies the correct resource name.

Invalid disk (*disk_no*) or partition(*partiton_no*) number specified. Error : *Error_code*

Invalid values specified for disk signature or partition number.

Recommended Action: See the corresponding error code for more information.

Cannot find the disk (*disk_no*) with signature *disk_signature* in the system.

Error : *Error_code*

The agent cannot find the specified disk with the given signature.

Recommended Action: See the corresponding error code for more information.

The specified disk (*disk_no*) with signature *disk_signature* is a dynamic disk.

Error : *Error_code*

The specified disk signature belongs to a dynamic disk, which is not supported by the agent.

Recommended Action: Verify that all disks configured using the Mount agent are basic disks. Verify the correct disk signatures are specified in the configuration definition of the agent. To configure dynamic disks, use the MountV agent.

Could not find the type of the child mount *mount_path*. Error : *Error_code*

The agent could not find the specified child mount

Recommended Action: Verify the configuration definition for the specified child mount. Make sure the specified mount exists.

Child mount *mount_path* is not a drive letter. Only one of level dependency is specified by the Mount agent. Mount resource at bottom should be a drive letter mount.

The agent is unable to resolve the specified drive letter.

Recommended Action: Verify the value specified for the MountResName attribute does not have another dependency of a mount path.

Could not find the type of the mount *mount_path*. Error : *Error_code*

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Failed to check whether the drive letter *drive_letter* is assigned or not. Error : *Error_code*

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

The specified drive letter *drive_letter* is a system drive.

Recommended Action: Specify a drive path other than the one for the system drive.

Child mount resource is not specified or configured

Incorrect value specified for the MountResName attribute.

Recommended Action: Verify that the configuration definition for the MountResName attribute specifies the correct resource name.

Invalid folder mount path (*mount_path*) specified

Recommended Action: Verify the value specified for folder mount is correct.

Invalid mount.

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Failed to gather Volumes Information on the system.Error : *Error_code*

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Failed to create the Volume object for DiskNo = *disk_no*, PartitionNo = *partition_no*.

Error : *Error_code*

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Failed to get the Volume name for DiskNo = *disk_no*, PartitionNo = *partition_no*.

Error : *Error_code*

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Failed to gather folder mount information for all the volumes. Error : *Error_code*. Volumes information could be incomplete.

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Failed to find the Volume for Volume Name = *volume_name*. Error : *Error_code*. Volumes information could be incomplete.

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Failed to find the folder mount information for Volume Name = *volume_name*.

Error : *Error_code*. Volumes information could be incomplete.

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

There is/are folder mount(s) mounted on the same drive. Folder mount = *folder_mount*.

Error :*Error_code*. Volumes information could be incomplete.

More than one folders are mounted on the same drive.

Recommended Action: Make sure no additional folders are mounted on the specified drive.

The configured Volume is also mounted as a drive letter. Failed to get the Drive letter.

Error : *Error_code*

Recommended Action: Unmount the volume or specify another volume information.

The configured Volume is also mounted as a drive letter *drive_letter*.

Recommended Action: Unmount the volume or specify another volume information.

Validate disk partition failed. Error : *Error_code*

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Could not find the Disk with number *disk_no*. Error : *Error_code*

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Unable to lock the volume. Disk No. = *disk_no*, PartitionNo = *partition_no*. Error : *Error_code*

The agent could not lock the volume while dismounting, indicating a potential hardware error or that other applications are accessing the volume.

Recommended Action: Verify that no other applications, including Windows Explorer, are accessing the volume. Verify the installed disk and the configured partition. Set the ListApplications attribute to 1 to obtain a list of applications that are accessing the volume.

Failed to unlock the volume. Disk No. = *disk_no*, PartitionNo =*partition_no*. Error : *Error_code*

The agent could not unlock the volume while dismounting, indicating a potential hardware error or that other applications are accessing the volume.

Recommended Action: Ensure that no other applications, including Windows Explorer, are accessing the volume. Verify the installed disk and the configured partition. Contact Veritas Technical Support if the problem persists.

Failed to mount the volume. Disk No. = *disk_no*, PartitionNo =*partition_no.* Error : *Error_code*

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Failed to dismount the volume. Disk No. = *disk_no*, PartitionNo =*partition_no*.

Error : *Error_code*

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Could not access the volume. Disk No. = *disk_no*, PartitionNo = *partition_no*

Mount Path = *Mount_path*. Error : *Error_code*

The specified device path is inaccessible, indicating a potential hardware error.

Recommended Action: Verify the installed disk and the configured partition. Also make sure no other application is accessing the volume. Set the ListApplications attribute to 1 to obtain a list of applications that are accessing the volume.

Failed to unmount the volume. Disk No. = *disk_no*, PartitionNo = *partition_no*

Mount Path = *Mount_path*. Error : *Error_code*

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

File system is dirty. Cannot online the resource. Unmounting the volume.

Error : *Error_code*

The filesystem at the specified driver letter or mount path is not clean.

Recommended Action: Set the AutoFSClean to 1 if you want the agent to clean the filesystem. Otherwise, clean the filesystem manually.

Drive letter *drive_letter* is not assigned Exit code : *Exit_code*

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Failed to flush the Volume. Error : *Error_code*

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Failed to get the application log information. Error : *Error_code*

The agent could not find the name of the application accessing the volume.

Recommended Action: See the corresponding error code for more information. Contact Veritas Technical Support if the problem persists.

Volumes information mismatched. Name1 = *vol1_name*, Name2 = *vol2_name*

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Unable to unmount the volume so cannot failover.

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Failed to add the administrative share. Error : *Error_code*

The agent could not assign the default administrative share for the configured drive letter.

Recommended Action: Verify that no administrative share for the drive letter exists.

Failed to delete the administrative share. Error : *Error_code*

The agent could not remove the administrative share assigned during the online operation.

Check disk failed. Cannot say properly online so unmounting the volume. Error : *Error_code*

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Failed to check whether the file system is dirty. Cannot online the resource. Unmounting the volume. Error : *Error_code*

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

The volume is already mounted as *drive_letter* (drive letter). Please correct the mount path. Error : *Error_code*

The target volume is mounted as a directory on an NTFS folder. The Mount agent does not support this configuration.

Please check whether the specified mount path is a removable media or network drive.

Recommended Action: Make sure the specified mount path is not a removable media or a network drive.

Unable to unmount the volume as there are applications that are still using the volume for WRITE. Please set ListApplication attribute to 1(if not set) to log the application names

The specified volume is being accessed by other applications.

Recommended Action: Set the ListApplications attribute to 1 to obtain a list of applications that are accessing the volume.

The volume (*vol_name*) is unmounted unexpectedly.

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

The base volume (*vol_name*) is not a NTFS volume

The specified mount point for the volume does not reside on an NTFS-formatted volume.

Recommended Action: Reset the attribute MountPath to point to an NTFS-formatted volume.

Mount path *Mount_path* does not exist

Invalid or incorrect mount path specified for the MountPath attribute.

Recommended Action: Verify that the configuration definition for the MountPath attribute specifies the correct mount path.

Failed to find the volume (*vol_name*) in the list. Error : *Error_code*

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Failed to find whether the base volume (*vol_name*) is NTFS or not. Error : *Error_code*

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Failed to get the volume name(GUID) for *guid*. Error : *Error_code*

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Error occurred while getting the drive letter(*drive_letter*) information.

This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Volume Manager (VMDg) agent error messages

The Agent is shutting down the system because it is configured so or it failed to start VxVM Service.

The VMDg agent is shutting down your system because you have provided SHUTDOWN as a value for VxVMFailAction attribute. This action is taken when the VMDg agent attempts to restart the Veritas Storage Agent service for the number of times specified in VxVMRestartAttempts attribute.

VCS license is not enabled in Volume Manager.

Storage Foundation was not installed with the HA license.

Recommended Action: Contact Veritas Technical Support.

Diskgroup not configured

No value specified for the DiskGroupName attribute.

Recommended Action: Verify the DiskGroupName attribute defined in the configuration file is assigned a value.

Could not initialize Volume Manager connection. Error = *Error_code*

The Veritas Enterprise Administrator Service is either not in a running state or is not responding to service request.

Recommended Action: Verify that the service is in a running state.

Could not get SCM handle [*Error Type:Error Code*]

The agent could not open a handle to the Service Control Manager.

Recommended Action: See the error type and error code for more information. Contact Veritas Technical Support if the problem persists.

Online diskgroup. Less than half of total disks available in the diskgroup.

Less than half the disks are available in the disk group.

Recommended Action: For the ForceImport to work, make sure at least 50% of the disks are available.

Online diskgroup : *Error_description*

The agent failed to bring the disk group online.

Recommended Action: See the corresponding error description for more information.

Offline diskgroup : *Error_description*

The agent failed to bring the disk group offline.

Recommended Action: See the corresponding error description for more information.

Init diskgroup : *Error_description*

The agent failed to initialize the disk group.

Recommended Action: See the corresponding error description for more information.

Failed to start VxOb Service (Error = *Error Type:Error Code*)

The agent failed to start the VxOb service.

Recommended Action: Try to start the VxOb service manually. See the corresponding error type and error code for more information.

Failed to query VxOb (Error = *Error Type:Error Code*)

The agent failed to query VxOb.

Recommended Action: Try to query VxOb manually. See the corresponding error type and error code for more information.

Failed to open VxOb (Error = *Error Type:Error Code*)

The agent failed to open VxOb.

Recommended Action: Try to open VxOb manually. See the corresponding error type and error code for more information.

Query imported cluster diskgroups : *Error_description*

Recommended Action: See the corresponding error description for more information.

Query SCSI Reservation : *Error_description*

Recommended Action: See the corresponding error description for more information.

One or more disks in this diskgroup are unsteady

One or more disks or volumes in the disk group are not in a healthy state.

Recommended Action: View the disks using the Veritas Enterprise Administrator console and verify the state of the disks.

Disk availability check : *Error_description*

Recommended Action: See the corresponding error description for more information.

Init diskgroup failed : The diskgroup and GUID do not match

Recommended Action: Use the vmgetdrive -details command to find the correct GUID.

MountV agent error messages

Could not initialize Volume Manager connection. Error = *Error_code*

The Veritas Enterprise Administrator Service is either not in a running state or is not responding to service request.

Recommended Action: Verify that the service is in a running state.

VCS license is not enabled in Volume Manager.

Storage Foundation was not installed with the HA license.

Recommended Action: Contact Veritas Technical Support.

***Attribute_name* not configured**

No value provided for the specified attribute.

Recommended Action: Verify the specified attribute defined in the configuration file is assigned a value.

Failed to get volume properties [*Error_type:Error_code*]

The agent failed to get the properties of the configured volume.

Recommended Action: See the corresponding error type and error code for more information.

Volume *folder_mount_path* is also mounted as Driveletter *drive_letter*

The specified volume is already mounted. The agent does not allow a volume to be mounted on multiple drives.

Recommended Action: Verify that the volume is not mounted at another drive letter. Also check that the volume is not mounted as an NTFS mount point.

Volume *drive_letter* is also mounted as NTFS Folder mount(s) at *folder_mount_path*

The agent does not allow a volume to be mounted on multiple drives.

Recommended Action: Verify that the volume is not mounted at a drive letter. Also check that the volume is not mounted as an NTFS mount point.

Volume *drive_letter* is Mounted as *mount_path* but configured as *volume_name*

The mount path for the volume specified in the configuration file is different from the actual mount path.

Recommended Action: Verify the mount path specified in the configuration file is correct.

Volume *volume_name* is already mounted as NTFS Folder *folder_mount_path*

The agent does not allow a volume to be mounted on multiple drives.

Recommended Action: Verify that the volume is not mounted at a drive letter. Also check that the volume is not mounted as an NTFS mount point.

Volume *volume_name* has been mounted multiple times (*no_of_times*)

The agent does not allow a volume to be mounted on multiple drives.

Recommended Action: Verify that the volume is not mounted at a drive letter. Also check that the volume is not mounted as an NTFS mount point.

Volume *volume_name* has also been mounted as (*drive_letter*)

The agent does not allow a volume to be mounted on multiple drives.

Recommended Action: Verify that the volume is not mounted at a drive letter. Also check that the volume is not mounted as an NTFS mount point.

Diskgroup not present

The specified disk group is not present on the system.

Recommended Action: Verify that the disk group exists on the system.

Volume not present

The specified volume is not present on the system.

Recommended Action: Verify that the volume exists on the system.

Connection to VM not initialized

The Veritas Enterprise Administrator Service is either not in a running state or is not responding to service request.

Recommended Action: Verify that the service is in a running state.

***Mount_path* is not suitable for folder mount [*Error_type:Error_code*]**

The specified MountPath is not suitable for a folder mount.

Recommended Action: See the description for the MountPath attribute for the VMDg agent. See the corresponding error type and error code for more information.

***Mount_path* is not on an NTFS formatted volume**

The specified MountPath is not an NTFS formatted volume.

Failed to unlock volume [*Error_type:Error_code*]

The agent could not unlock the volume while dismounting, indicating a potential hardware error or that other applications are accessing the volume.

Recommended Action: Ensure that no other applications, including Windows Explorer, are accessing the volume. Verify the installed disk and the configured partition. Contact Veritas Technical Support if the problem persists.

Failed to lock volume [*Error_type:Error_code*]

The agent could not lock the volume while dismounting, indicating a potential hardware error or that other applications are accessing the volume.

Recommended Action: Verify that no other applications, including Windows Explorer, are accessing the volume. Verify the installed disk and the configured partition. Set the ListApplications attribute to 1 to obtain a list of applications that are accessing the volume.

Filesystem at *drive_letter/mount_path* is not clean [*Error_type:Error_code*]

The file system at the specified drive letter or mount path is not clean

Recommended Action: Set the AutoFSClean to 1 if you want the agent to clean the filesystem. Otherwise, clean the filesystem manually.

Failed to add administrative share

The agent could not assign the default administrative share. An administrative share is already added for this volume.

Failed to delete administrative share [*Error_type:Error_code*]

The agent could not remove the administrative share assigned during the online operation.

Recommended Action: See the corresponding error type and error code for more information.

Troubleshooting network agents

Network agents comprise the NIC agent, IP agent, and Lanman agent.

NIC agent error messages

Initialization failed

The agent failed to retrieve network information. This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Invalid MAC Address format

Incorrect format for MAC Address specified in the MACAddress attribute.

Recommended Action: Verify the MAC Address is in the format AA-BB-CC-DD.

Invalid MAC Address (*MAC_Address*) specified.

Invalid value specified for adapters physical address.

Recommended Action: Verify the MACAddress attribute defined in the configuration file is assigned a value.

Adapter (*Adapter_Name*) Connection status cannot be found

The agent failed to retrieve the connection status of the NIC.

Recommended Action: Contact Veritas Technical Support.

PingHostList not specified

No value is specified for PingHostList attribute even though the UseConnectionStatus is set to False.

Recommended Action: Specify a value for PingHostList.

UDP echo attempt failed

This is a network error.

Recommended Action: Contact Veritas Technical Support.

UDP check failed

This is a network error.

Recommended Action: Contact Veritas Technical Support.

Failed to clear all IPs from adapter *Adapter_Name*

The agent failed to deallocate all IPs assigned to a network card. The agent does so to avoid IP conflict.

Recommended Action: Contact Veritas Technical Support

Adapter (*Adapter_Name*) not found

The agent could not detect the specified adapter.

Recommended Action: Verify the MACAddress attribute defined in the configuration file is assigned a value.

Adapter (*Adapter_Name*) is not connected

The adapter specified in the MACAddress attribute is not available on the network.

Recommended Action: Check the network connections. If the problem persists, verify the MACAddress attribute defined in the configuration file is assigned a value.

Failed to get IFEntry for adapter *Adapter_Name*

The agent failed to retrieve an entry for the specified adapter from the IF table. This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

TCP/IP is not enabled for adapter (*Adapter_Name*)

TCP/IP is not enabled for the specified adapter.

Recommended Action: Enable TCP/IP for the adapter.

Error percentage exceeded for (*Adapter_name*)

The adapter has exceeded the error percentage for sending packets, specified in the MaxTxErrorPercentage attribute.

Recommended Action: Contact Veritas Technical Support.

Error percentage exceeded limit

The adapter has exceeded the error percentage, specified by MaxTxErrorPercentage, by the number of times specified in MaxTxErrInterval attribute.

Recommended Action: Contact Veritas Technical Support.

IP agent error messages

Initialization failed

The agent failed to retrieve network information. This is a VCS internal error.

Recommended Action: Contact Veritas Technical Support.

Invalid attributes

Invalid values specified for the agent attributes.

Recommended Action: Verify the attributes defined in the configuration file are assigned a correct value.

Invalid MacAddress format

Incorrect format for MAC Address specified in the MACAddress attribute.

Recommended Action: Verify the MAC Address is in the format AA-BB-CC-DD.

Invalid IP format

Incorrect format for IP address specified in the Address attribute.

Recommended Action: Verify the IP address is in the correct format.

Ping failed for IP (*IP_Address*)

The agent failed to ping the specified IP address.

Recommended Action: Verify the specified IP address exists in the network. Check the network connections if the problem persists.

Host *IP_Address* already exists

The specified IP address already exists in the network

Recommended Action: Verify the value specified for Address attribute is unique.

IP (*IP_Address*) is Admin IP

The specified virtual IP is the base ID for a NIC in the network.

Recommended Action: Specify a unique IP address.

Incorrect IP (*IP_Address*)

Invalid value specified for the Address attribute.

Recommended Action: Verify the Address attribute defined in the configuration file is assigned a correct value.

Incorrect subnet mask (*SubNet_Mask*)

Invalid value specified for the SubNetMask attribute.

Recommended Action: Verify the SubNetMask attribute defined in the configuration file is assigned a correct value.

Adapter (*IP_Address*) not found

The agent could not detect the specified adapter in the network.

Recommended Action: Verify the Address attribute defined in the configuration file is assigned a correct value.

DHCP is enabled on Adapter (*IP_Address*)

DHCP is enabled on the specified adapter.

Recommended Action: Disable DHCP and specify an IP address for the adapter.

Invalid Adapter (*IP_Address*)

Invalid value specified for the Address attribute.

Recommended Action: Verify the Address attribute defined in the configuration file is assigned a correct value.

Invalid MacAddress (*MAC_Address*) specified

Invalid value specified for the MACAddress attribute.

Recommended Action: Verify the MACAddress attribute defined in the configuration file is assigned a correct value.

TCP/IP is not enabled for adapter (*Adapter_Name*)

TCP/IP is not enabled for the specified adapter.

Recommended Action: Enable TCP/IP for the adapter.

Lanman agent error messages

Virtual server name specified in the configuration is invalid (*Virtual Server Name*)

The virtual server name exceeds the limit of 16 characters, or it contains invalid characters.

Recommended Action: Verify that the NetBios name is less than sixteen characters, and that is does not contain any of the following characters: /\:*?"<>|=.

Virtual name already exists in the network

The specified virtual name already exists in the network.

Recommended Action: Verify the VirtualName attribute defined in the configuration file is assigned a unique value.

Failed to update DNS entry (error_type:*Error Type*, error_code:*Error Code*)

The agent failed to update the DNS entry.

Recommended Action: Verify the DNS server is alive, and the reverse lookup zone is configured.

Failed to update Computer account in Active directory (error_type: *Error Type*, error_code: *Error Code*)

The agent failed to add the computer account to the Active Directory.

Recommended Action: Verify whether the user, in whose context the VCS Helper service runs, has the required permissions.

Failed to register Virtual server name to Netbios (error_type:*Error type*, error_code: *Error Code*)

The agent failed to register the virtual server name with the NetBios.

Recommended Action: Verify whether the virtual name is already configured. Run nbtstat - n to check NetBios bindings. Contact Veritas Technical Support if the problem persists.

Failed to start the 'Veritas Cluster Server Helper' service. (error_type: *Error Type*, error_code:*Error Code*)

The agent failed to start the specified service.

Recommended Action: Verify whether the service is installed on the node. Also verify whether the user, in whose context the service will run, has the required permissions.

IP address specified for MultiNetInfo is invalid (entry=*IP Address*)

Invalid IP address specified in the MultiNetInfo attribute.

Recommended Action: Verify the MultiNetInfo attribute defined in the configuration file is assigned correct values.

SubnetMask specified for MultiNetInfo is invalid (entry=*subnet_mask*)

Invalid subnet mask specified in the MultiNetInfo attribute.

Recommended Action: Verify the MultiNetInfo attribute defined in the configuration file is assigned correct values.

WINS address specified for MultinetInfo is invalid (entry=*wins_address*)

Invalid WINS address specified in the MultiNetInfo attribute.

Recommended Action: Verify the MultiNetInfo attribute defined in the configuration file is assigned correct values.

VirtualName attribute is not specified in the configuration

No value specified for VirtualName attribute.

Recommended Action: Verify the VirtualName attribute defined in the configuration file is assigned a value.

IP address specified in the configuration is invalid

Invalid value specified for the IP address.

Recommended Action: Verify the value specified for the IP address is valid.

SubnetMask specified in the configuration is invalid

Invalid value specified for the subnet mask.

Recommended Action: Verify the value specified for the subnet mask is valid.

MultinetInfo attribute specified in the configuration is in invalid format

The value specified for MultiNetInfo attribute is not in the standard format.

Recommended Action: Verify the format for the MultiNetInfo attribute.

There are no valid IP addresses specified in the configuration

All the IP addresses specified in the configuration are invalid.

Recommended Action: Verify the IP addresses specified in the configuration file.

IP address specified in the configuration is not online. (ip= *IP Address*)

The specified IP address is not online in the cluster.

Recommended Action: Verify the specified IP address exists in the network.

Mismatch in the subnet mask specified in configuration and the actual subnet mask configured for the adapter

The subnet mask specified in the configuration file does not match with the subnet mask for the configured adapter.

Recommended Action: Verify the subnet mask specified in the MultiNetInfo attribute is correct.

WINS address specified in the configuration is not found in the adapter's WINS address list

The specified WINS address does not exist in the WINS address list of the adapter.

Recommended Action: Verify the WINS address specified in the MultiNetInfo attribute is correct.

Attempt to start 'Veritas Cluster Server Helper' service has failed. (error_type: *Error Type*, error_code: *Error Code*)

The agent failed to start the specified service.

Recommended Action: Verify whether the service is installed on the node. Also verify whether the user, in whose context the service will run, has the required permissions.

Attempt to stop 'Veritas Cluster Server Helper' service has failed. (error_type: *Error Type*, error_code: *Error Code*)

The agent failed to stop the specified service.

Recommended Action: Verify whether the service is installed on the node. Also verify whether the user, in whose context the service will run, has the required permissions.

Attempt to online the Lanman resource has failed. (error_type: *Error Type*, error_code: *Error Code*)

The agent failed to online the Lanman resource. This is a VCS internal error.

Recommended Action: See the Windows error code for more information. Contact Veritas Technical Support if the problem persists.

V-16-10051-3051 No DNS servers IPv6 address found. Cannot update DNS. Please provide IPv6 addresses for the DNS servers either on the NIC or the Lanman configuration.

This error is specific to an IPv6 environment. The Lanman agent has failed to update the virtual IPv6 address entry in the DNS records as the DNS server details are not specified.

The Lanman resource itself is able to come online, but other resources in the service group that depend on the Lanman resource (FileShare for example) may fail to come online as the virtual server name that uses an IPv6 address is not updated in the DNS.

In an IPv6 environment, the Lanman agent relies on the DNS records to validate the server name on the network.

Recommended Action: If the virtual IP is an IPv6 address then you must specify the DNS server details, either in the network adapter properties or in the Lanman agent's AdditionalDNSServers attribute.

Troubleshooting FileShare agents

FileShare agents comprise the FileShare agent and the CompositeFileShare agent.

FileShare agent error messages

Attribute value(s) invalid for *attr_name*

The specified attribute name has invalid value.

Recommended Action: Verify the configuration file and make sure the correct attribute value is specified.

Failed to open folder *folder_name* [*ErrorType:Error Code*]

The agent failed to open the folder specified in the ShareName attribute.

Recommended Action: Verify whether the specified folder exists. Check the permissions for the folder.

Folder *folder_name* (*system_nameshare_name*) already shared**

The specified folder name is already shared.

Recommended Action: Verify the configuration file for the specified ShareName.

Failed to share folder *folder_name* as (*system_nameshare_name*) [*ErrorType:Error Code*]**

The agent failed to share the specified folder.

Recommended Action: Verify the configuration file for the specified ShareName.

MaxUsers for share *share_name* is less than configured

The number of users authorized to access a share is less than the number of users specified in the MaxUsers attribute.

Recommended Action: Make sure the number of users authorized to access a share and the number of users specified in the MaxUsers attribute is same.

Failed to unshare folder *folder_name* as (*system_nameshare_name*) [*ErrorType:Error Code*]**

The agent failed to unshare the specified folder.

Recommended Action: Verify the configuration file for the specified share name.

Path mismatch for share *share_name*

The specified path name is already shared.

Recommended Action: Verify that the share name does not already exist on the network. If the share does not exist, contact Veritas Technical Support.

System error occurred for folder *folder_name* Error [*Error_code*]

Recommended Action: See the Windows error code for more information. Contact Veritas Technical Support if the problem persists.

FileShare Error : The Server service is not available

The agent requires the Server service to be started before it can bring the resource online.

Recommended Action: Verify that the Server service is running properly.

Permissions error occurred for folder *folder_name* Error [*error_id*]

Recommended Action: See the corresponding error_id for more information. Error ID 6 denotes a mismatch in the configured permission and the actual permission for the folder share. Error ID 7 denotes an error while matching the configured permission and actual permission for the share.

Share name *share_name* contains invalid characters

The specified share name contains invalid characters.

Recommended Action: Make sure the specified share name does not contain special characters (\, /, :, *, ?, <, >, |).

CompositeFileShare agent error messages

Attribute value(s) invalid for *attr_name*

The specified attribute name has an invalid value.

Recommended Action: Verify the configuration file and make sure the correct attribute value is specified.

Failed to open folder *folder_name* [*ErrorType:Error Code*]

The agent failed to open the folder specified in the ShareName attribute.

Recommended Action: Verify whether the specified folder exists. Check the permissions for the folder.

Folder *folder_name* (*system_nameshare_name*) already shared**

The specified folder name is already shared.

Recommended Action: Verify the configuration file for the specified ShareName.

Failed to share folder *folder_name* as (*system_nameshare_name*) [*ErrorType:Error Code*]**

The agent failed to share the specified folder.

Recommended Action: Verify the configuration file for the specified ShareName.

MaxUsers for share *share_name* is less than configured

The number of users authorized to access a share is less than the number of users specified in the MaxUsers attribute.

Recommended Action: Make sure the number of users authorized to access a share and the number of users specified in the MaxUsers attribute is same.

Failed to unshare folder *folder_name* as (*system_nameshare_name*) [*ErrorType:Error Code*]**

The agent failed to unshare the specified folder.

Recommended Action: Verify the configuration file for the specified share name.

Path mismatch for share *share_name*

The specified path name is already shared.

Recommended Action: Verify that the share name does not already exist on the network. If the share does not exist, contact Veritas Technical Support.

Composite FileShare Error : The Server service is not available

The agent requires the Server service to be started before it can bring the resource online.

Recommended Action: Verify that the Server service is running properly.

Permissions error occurred for folder *folder_name* Error [*error_id*]

Recommended Action: See the corresponding error_id for more information. Error ID 6 denotes a mismatch in the configured permission and the actual permission for the folder share. Error ID 7 denotes an error while matching the configured permission and actual permission for the share.

Share name *share_name* contains invalid characters

The specified share name contains invalid characters.

Recommended Action: Make sure the specified share name does not contain special characters (\, /, :, *, ?, <, >, |).

Troubleshooting services and application agents

Services and application agents comprise GenericService, ServiceMonitor, and Process agent.

IIS agent error messages

Failed to create the IIS services map.

Recommended Action: Verify the SiteType attribute is defined correctly in the configuration. See the associated Windows error type and error code for more information.

Failed to parse the argument list.

The agent could not pass all arguments associated with the agent's resource type definition.

Recommended Action: Verify the configuration definition of the agent.

Failed to allocate memory for IIS Service resource object.

Recommended Action: Verify the system has adequate memory to run IIS and VCS.

Failed to find the service object. Please check the SiteType attribute.

Recommended Action: Verify the configuration definition of the SiteType attribute.

Arguments are not specified correctly, Please check ArgList, *ResName*

Some attributes associated with the agent are not defined correctly.

Recommended Action: Verify the configuration definition of the specified attribute.

Configuration error. Attribute *attribute* is not configured.

Recommended Action: Verify the configuration definition of the specified attribute.

Error Type.* Internal error IP (%s) Failed to convert IP address to ASCII. *Error Code

Recommended Action: Verify the configuration definition of the IP resources. See the associated Windows error type and error code for more information.

Failed to start/stop the application pool.

Recommended Action: Verify the name of the application pool is correctly defined in the agent's configuration.

Failed to get the application pool state. *Error Type. Error Code*.

The agent could not determine the state of the configured application pool.

Recommended Action: Verify the name of the application pool is correctly defined in the agent's configuration. See the associated Windows error type and error code for more information.

Failed to get the IIS virtual directory object *object_name*. *Error Type. Error Code*.

The agent could not access the specified virtual directory object.

Recommended Action: Verify the virtual directory object for the site exists. See the associated Windows error type and error code for more information.

Failed to start the application pool *poolname*. *Error Type. Error Code*.

The agent could not start the specified application pool.

Recommended Action: Verify the configuration definition of the application pool.

Failed to get the application pool interface. *Error Type. Error Code*.

The agent could not access the COM interfaces for the application pool.

Recommended Action: See the associated Windows error type and error code for more information.

Failed to stop the application pool *poolname*. *Error Type. Error Code*.

The agent could not stop the specified application pool.

Recommended Action: Verify the configuration definition of the application pool.

Failed to start the IIS site/virtual server. *Error Type. Error Code*.

The agent could not start the specified IIS site or virtual server.

Recommended Action: Verify the specified site or server exists and verify its configuration definition.

Failed to stop the IIS site/virtual server. *Error Type. Error Code*.

The agent could not stop the specified IIS site or virtual server.

Recommended Action: Verify configuration definition of the specified site or server.

Failed to determine the state of the IIS site/virtual server. *Error Type. Error Code*.

The agent could not determine the state of the specified IIS site or virtual server.

Recommended Action: Verify configuration definition of the specified site or server.

Failed to get the application pool state and the bind path. *Error Type. Error Code*.

The agent could not determine the state of the specified application pool.

Recommended Action: Verify configuration definition of the specified application pool.

Failed to get the IIS version. *Error Type. Error Code*.

The agent could retrieve the version information for IIS.

Recommended Action: Verify that a supported IIS version is installed on all nodes in the cluster. See the associated Windows error type and error code for more information.

Failed to get the registry *keyname*. *Error Type. Error Code*.

The agent could not access the specified registry key.

Recommended Action: Verify the specified registry key exists.

Failed to start service *servicename*. *Error Type. Error Code*.

The agent could not start the specified service.

Recommended Action: Verify the service exists and is correctly defined in the configuration file. Verify that the service is supported by the host.

Failed to get the state of service *servicename. Error Type. Error Code*.

The agent could not determine the state of the specified service.

Recommended Action: Verify the service exists and is correctly defined in the configuration file. Verify that the service is supported by the host

Failed to stop service *servicename. Error Type. Error Code*.

The agent could not stop the specified service.

Recommended Action: Verify the service exists and is correctly defined in the configuration file. Verify that the service is supported by the host

Failed to initialize service object *servicename. Error Type. Error Code*.

The agent could not initialize the specified service.

Recommended Action: Verify the service exists and is correctly defined in the configuration file. Verify that the service is supported by the host

GenericService agent error messages

The ServiceName attribute has not been configured

No value specified for the ServiceName attribute.

Recommended Action: Verify the ServiceName attribute defined in the configuration file is assigned a value.

The password attribute has not been configured

No value specified for the Password attribute.

Recommended Action: Verify the Password attribute defined in the configuration file is assigned a value.

Failed to get the password attribute. Error = *Error Code*

The agent failed to decrypt the specified password.

Recommended Action: Use the vcsencrypt utility to get an encrypted password. Specify the encrypted password to the Password attribute in the configuration file.

Failed to convert the password attribute. Error = *Error Code*

The agent failed to convert the password attribute. This is a VCS internal error.

Recommended Action: Offline and online the service group. Contact Veritas Technical Support if the problem persists.

Failed to allocate memory

The agent failed to allocate the memory. This is a VCS internal error.

Recommended Action: Offline and online the service group. Contact Veritas Technical Support if the problem persists.

Failed to lookup the account name '*Account_Name*'. Error = *Error Code*

The agent failed to obtain the SID for the specified user.

Recommended Action: Verify that the domain controller is accessible, or the specified user exists. See the error code if the problem persists.

Failed to open the SCM handle. Error = *Error Code*

The agent failed to obtain the SCM handle. This occurs if the specified Service Control Manager (SCM) database does not exist, or the requested access is denied.

Recommended Action: Verify that SCM can be run on the host.

The service '*Service_Name*' is not in stopped state

The agent failed to online the service because the service was not in STOPPED state.

Recommended Action: Verify the state of the service and re-issue the online command.

Failed to decrypt the password

The agent failed to decrypt the specified password.

Recommended Action: Verify that it the password is encrypted using the VCS Encrypt utility (described in the *Cluster Server Administrator's Guide*.).

Failed to change the user context of the service '*Service_Name*' to '*Domain\NameUsername*.Error = *Error Code*

The agent failed to change the user context in which the service will run.

Recommended Action: See the Windows error code for further information.

Failed to start the service '*Service_Name*'. Error = *Error Code*

The agent failed to start the specified service.

Recommended Action: Verify the binaries for the specified service exist, or the service is configured to run in the context of the specified user. Start the service manually if the problem persists.

The service '*Service_Name*' did not start within the specified time limit

The agent failed to online the service within the time specified in the DelayAfterOnline attribute.

Recommended Action: Verify the state of the service in the SCM. If the state of the service is START PENDING, increase the time specified in DelayAfterOnline attribute. If the problem persists, start the service manually.

Failed to wait for the service '*Service_Name*' to start. Error = *Error Code*

The agent did not wait for the service to start. This is a VCS internal error. Refer to the documentation specific to the service to determine the cause.

Failed to get the user context of the running service '*Service Name*'. Error = *Error Code*

The agent failed to obtain the information about the user in whose context, the specified service will run.

Recommended Action: Verify that the user exists, or the domain controller is accessible. See the error code if the problem persists.

Failed to get the SID for user '*DomainNameUsername*'**

The agent failed to obtain the SID for the specified user.

Recommended Action: Verify that the domain controller is accessible, or the specified user exists. See the error code if the problem persists.

The service '*Service_Name*' is not running under the context of user *DomainNameUsername***

The user information specified in the UserAccount information is different from the user information, in whose context the service will run.

Recommended Action: Stop the service and modify the user information. Re-issue the online command.

The service '*Service_Name*' shares the same process with other services. Killing it might affect those services

If the service has an invalid state, the agent attempts to offline the service. The agent fails to offline the service, if the process is shared with other services. In such cases, the agent attempts to kill the shared process. You risk potential data loss.

Failed to enable the '*Privilege_Name*' privilege. Error = *Error Code*

The agent failed to enable the privilege required to kill a process. See the Windows error code for further information.

Failed to open the service '*Service_Name*'. Error = *Error Code*

The resource definition in the configuration file contains an invalid, or incorrect service name.

Recommended Action: Verify the ServiceName attribute defined in the configuration file is the same as the display name used by SCM. Verify that the specified service was opened successfully.

Failed to query the status of the service 'Service Name'. Error = *Error Code*

The agent failed to determine the status of the service. This is a VCS internal error.

Recommended Action: Offline and online the service group. Contact Veritas Technical Support if the problem persists.

The service '*Service_Name*' is not in running state. Attempt to stop it might be unsuccessful.

The agent failed to offline the service because the service is not in STARTED state.

Recommended Action: Verify the state of the service and re-issue the online command.

The service '*Service_Name*' did not stop. Error = *Error Code*.

The agent failed to stop the specified service.

Recommended Action: Verify the binaries for the specified service exist, or the service is configured to run in the context of the specified user. Stop the service manually if the problem persists.

The service '*Service_Name*' did not stop within the specified timeout. Error = *Error Code*

The agent failed to offline the service within the time specified in the DelayAfterOffline attribute.

Recommended Action: Verify the state of the service in the SCM. If the state of the service is STOP PENDING, increase the time specified in DelayAfterOffline. If the problem persists, stop the service manually.

ServiceMonitor agent error messages

The ServiceOrScriptName attribute has not been configured

No value specified for the ServiceorScriptName attribute.

Recommended Action: Verify the ServiceorScriptName attribute defined in the configuration file is assigned a value.

Failed to lookup the account name '*Account_Name*'. Error = *Error Code*

The agent failed to obtain the SID for the specified user.

Recommended Action: Verify that the domain controller is accessible, or the specified user exists. See the error code if the problem persists.

The password attribute has not been configured

No value specified for the Password attribute.

Recommended Action: Verify the Password attribute defined in the configuration file is assigned a value and contains alphabets only.

Failed to get the password attribute. Error = *Error Code*

The agent failed to decrypt the specified password.

Recommended Action: Verify that the password is encrypted using the VCS Encrypt utility (described in the *Cluster Server Administrator's Guide*).

Failed to convert the password attribute. Error = *Error Code*

The agent failed to convert the password attribute. This is a VCS internal error.

Recommended Action: Offline and online the service group. Contact Veritas Technical Support if the problem persists.

Failed to open the SCM handle. Error = *Error Code*

The agent failed to obtain the SCM handle. This occurs if the SCM database does not exist, or if the SCM manager denies the requested access.

Recommended Action: Verify that SCM can be run on the host.

Failed to open the service '*Service_Name*'. Error = *Error Code*

The resource definition in the configuration file contains an invalid, or incorrect service name.

Recommended Action: Verify that the ServiceOrScriptName defined in the configuration file is the same as the display name used by SCM. Also verify that the specified service is supported by the SCM version on the host and that the specified service was opened successfully.

Failed to query the status of the service '*Service_Name*'. Error = *Error Code*

The agent failed to determine the status of the service. This is a VCS internal error.

Recommended Action: Verify that the attribute ServiceOrScriptName defined in the configuration file is the same as the display name used by SCM. Otherwise, offline and online the service group. Contact Veritas Technical Support if the problem persists.

The service '*Service_Name*' is not in running state

The agent failed to offline the service because the service is not in STARTED state.

Recommended Action: Verify the state of the service and re-issue the online command.

Failed to get the user context of the running service '*Service_Name*'. Error = *Error Code*

The agent failed to obtain the information about the user in whose context, the specified service will run.

Recommended Action: Verify that the user exists, or the domain controller is accessible. See the error code if the problem persists.

Failed to get the SID for user '*DomainNameUsername*'**

The agent failed to obtain the SID for the specified user.

Recommended Action: Verify that the domain controller is accessible, or the specified user exists. See the error code if the problem persists.

The service '*Service_Name*' is not running under the context of user *DomainNameUsername***

The user information specified in the UserAccount information is different from the user information, in whose context the service will run.

Recommended Action: Stop the service and modify the user information. Re-issue the online command.

Failed to launch the monitor script '*Script_Name*'. Error = *Error Code*

The agent failed to launch the script to monitor the service.

Recommended Action: Verify that the script is accessible. See the Windows error code if the problem persists.

Monitor script has not responded within the specified timeout of *Timeout period* seconds. Error = *Error Code*

The agent failed to receive a response from the monitor script.

Recommended Action: Review the configuration definition of the attribute MonitorProgTimeout. Increase its value if required.

Failed to terminate the monitor program. Error = *Error Code*

The agent failed to terminate the script specified in the *MonitorService* attribute.

Recommended Action: Stop the agent. See the Windows error code, if the problem persists.

Failed to get exit code of the monitor program. Error = *Error Code*

The agent failed to get a response from the script that monitors a specified service.

Recommended Action: Verify that the exit codes returned by the script conform to VCS standards. See the Windows error code for further information. Contact Veritas Technical Support if the problem persists.

Failed to enable the *Privilege_Name*' privilege. Error = *Error Code*

The agent failed to enable the privilege required to kill a process. See the Windows error code for further information.

Invalid Service name [service_name].

The resource definition in the configuration file contains an invalid or incorrect service name.

Recommended Action: Verify that the attribute ServiceOrScriptName defined in the configuration file is the same as the display name used by SCM. Also verify that the specified service is supported by the SCM version on the host.

Process agent error messages

Failed to convert the argument list. Error = *Error Code*

The agent failed to convert the argument list. This is a VCS internal error.

Recommended Action: Offline and online the service group. Contact Veritas Technical Support if the problem persists.

Failed to lookup the account name for the user '*Account_Name*'. Error = *Error Code*

The agent failed to obtain the SID for the specified user.

Recommended Action: Verify that the domain controller is accessible, or the specified user exists. See the error code if the problem persists.

The password attribute has not been configured

No value specified for the Password attribute.

Recommended Action: Verify the Password attribute defined in the configuration file is assigned a value and contains alphabets only. If the problem persists, verify that the password is encrypted using the VCS Encrypt utility (described in the *Cluster Server Administrator's Guide*).

Failed to get the password attribute. Error = *Error Code*

The agent failed to decrypt the specified password.

Recommended Action: Use the vcsencrypt utility to get an encrypted password. Specify the encrypted password to the Password attribute in the configuration file.

Failed to convert the password attribute. Error = *Error Code*

The agent failed to convert the password attribute. This is a VCS internal error.

Recommended Action: Offline and online the service group. Contact Veritas Technical Support if the problem persists.

StartProgram has not been configured

No value specified for the StartProgram attribute.

Recommended Action: Verify the StartProgram attribute defined in the configuration file is assigned a value.

The process (*Process ID*) is not alive. Error = *Error Code*

The process specified in the StartProgam attribute is not alive.

Recommended Action: Verify the StartProgram attribute defined in the configuration file is assigned a value. See the Windows error code if the problem persists.

Failed to get cmd line of the process(*Process_ID*). Error= *Error Code*

The agent failed to determine the command line string for the process mentioned in the StartProgram attribute.

Recommended Action: Verify the StartProgram attribute defined in the configuration file is assigned a value. See the Windows error code if the problem persists.

The command line of the process(*Process_ID*) does not match with that configured in the attributes

The command line of the process, determined by the agent, did not match with the command line specified in the StartProgram attribute.

Recommended Action: Verify the StartProgram attribute defined in the configuration file is assigned a correct value.

Failed to get the user context of the process(*Process_ID*). Error= *Error Code*

The agent failed to obtain the information about the user in whose context, the specified process will run.

Recommended Action: Verify that the user exists, or the domain controller is accessible. See the error code if the problem persists.

The user context of the process (*Process_ID*) does not match with that configured in the attributes

The user context of the process, determined by the agent, did not match with the user context specified in the UserName attribute.

Recommended Action: Verify the UserName attribute defined in the configuration file is assigned a correct value.

Failed to determine whether the process is running or not. Error = *Error Code*

The agent failed to determine whether the process specified in the StartProgram attribute is running.

Recommended Action: Verify the StartProgram attribute defined in the configuration file is assigned a value. See the Windows error code if the problem persists.

Failed to enable '*Privilege_Name*' privilege. Error = *Error Code*

The agent failed to enable the privilege required to kill a process. See the Windows error code for further information.

Failed to launch the program '*Program_Name*'. Error = *Error Code*

The agent failed to launch the program specified in the StartProgram, CleanProgram, MonitorProgram, or StopProgram attributes.

Recommended Action: Verify the attributes defined in the configuration file are assigned a value. See the Windows error code if the problem persists.

Failed to get command line of the process (*Process_ID*). Error= *Error Code*

The agent failed to determine the command line of the process.

Recommended Action: Verify the attributes defined in the configuration file are assigned a value. See the Windows error code if the problem persists.

Failed to wait for the process (*Process_ID*) to stop. Error = *Error Code*. Killing it

The process specified in the StartProgram, CleanProgram, MonitorProgram, or StartProgram attribute did not respond in the specified timeout. The agent will try to kill the process.

Recommended Action: Increase the timeout specified in the MonitorProgamTimeout attribute. See the Windows error code if the problem persists.

Failed to get the ACL information. Error = *Error Code*

The agent failed to retrieve the Access Control List of the interactive desktop and window station.

Recommended Action: See the Windows error code for more information. Contact Veritas Technical Support if the problem persists.

Failed to get the ACE. Error = *Error Code*

The agent failed to retrieve the Access Control Entry for the ACL of the interactive desktop and window station.

Recommended Action: See the Windows error code for more information. Contact Veritas Technical Support if the problem persists.

Failed to add an ACE entry. Error = *Error Code*

The agent failed to add an ACE to the ACL of the interactive desktop and window station.

Recommended Action: See the Windows error code for more information. Contact Veritas Technical Support if the problem persists.

MSMQ agent error messages

Clustered MSMQ service is NOT bound to correct port.

You might encounter this error in one or more of the following conditions:

- If a DNS lag occurs in the network
  The MSMQ agent verifies that the clustered MSMQ service is bound to the correct virtual IP and port. By default, the agent performs this check only once

during the Online operation. If the clustered MSMQ service is not bound to the correct virtual IP and port, the agent stops the service and the resource faults.
If a DNS lag occurs in the network, checking the port bindings once might not reflect the accurate and most recent status.
Recommended Action: Configure the number of times that the MSMQ agent checks for the correct network binding.
Create the `VirtualIPPortCheckRetryCount` tunable parameter under the following registry key:

`HKEY_LOCAL_MACHINE\SOFTWARE\VERITAS\VCS\BundledAgents\MSMQ`

Set the `DWORD` value of this parameter to an integer greater than 1.
For more information, refer to the *Cluster Server Administrator's Guide*.

- The default MSMQ service is not bound to the correct IP and port
  If the network bindings for the default MSMQ service are incorrect, the clustered MSMQ resource fails to come online on a Windows Server 2012 or 2012 R2 node.
  Recommended Action: Create the `BindInterfaceIP` registry key for the default MSMQ service, restart the service, and bring the clustered MSMQ resource or the service group online.

  For more information, refer to the following:

  - Microsoft Knowledge Base Article 974813

    Note: For a Windows Server 2012 or 2012 R2 system, you do not need to install the hotfix mentioned in the article.

  - *Cluster Server Administrator's Guide*

Troubleshooting infrastructure and support agents

Infrastructure and Support agents comprise NotifierMngr agent, Registry Replication agent, Proxy agent, Phantom agent, FileNone agent, ElifNone agent, FileOnOff agent, and FileOnOnly agent.

Notifier Manager agent error messages

Failed to Start Notifier Process *Process_ID*. (Windows Error Code = *Error Code*)

The agent failed to start the specified notifier process. This could occur due to an incorrect path name specified in the configuration, low memory on the system, or a Windows-specific error, defined by the corresponding error code.

Recommended Action: Verify the configuration definition of the agent. Verify that the system has adequate memory.

Failed to Open Notifier Process. PID = *Process_ID* (Windows Error Code = *Error Code*)

The notifier process was terminated.

Recommended Action: VCS marks the resource as faulted. Wait, then try to bring the resource online again.

Failed to Offline Notifier Process. PID = *Process_ID* (Windows Error Code = *Error Code*)

The agent could not terminate the notifier process. This indicates a Windows-specific error, defined by the corresponding error code.

Recommended Action: Try terminating the process manually using Task Manager.

Failed to Clean Notifier Process. PID = *Process_ID* (Windows Error Code = *Error Code*)

The agent could not terminate the notifier process. This indicates a Windows-specific error, defined by the corresponding error code.

Recommended Action: Try terminating the process manually using Task Manager.

Invalid SNMP severity specified

Invalid severity level specified for SNMP notifications,

Recommended Action: Verify the severity level specified in the SnmpConsoles attribute has "Warning", "Information", "Error", or "SevereError" as the severity level.

Invalid SMTP severity specified

Invalid severity level specified for SMTP notifications,

Recommended Action: Verify the severity level specified in the SmtpRecipients attribute has "Warning", "Information", "Error", or "SevereError" as the severity level.

Expected correct SNMP and | or SMTP options

Invalid values specified for SNMP and SMTP attributes.

Recommended Action: Verify the attributes defined in the configuration file are assigned valid values.

Unable to online resource.Please specify correct SNMP and | or SMTP options

Invalid values specified for SNMP and SMTP attributes.

Recommended Action: Verify the attributes defined in the configuration file are assigned valid values.

Unable to find VCS_HOME. Error Code = *Error Code*

The value for VCS_HOME environment variable is incorrect.

Recommended Action: Verify the value for the VCS_HOME environment variable. Contact Veritas Technical Support if the problem persists.

Registry Replication agent error messages

Directory does not exist (or) could not create (directory=*Drive name:path*)

The registry replication directory specified in the configuration either does not exist or could not be created.

Recommended Action: Verify that the shared drive is accessible, and that no file in that drive has the same name as that of the ReplicationDirectory attribute. Also, verify whether the required permissions are available to create the directory.

The name of the local host is invalid (hostname= *host_name*)

This is a VCS internal error.

Recommended Action: Verify that the local host has a valid name. Contact Veritas Technical Support if the problem persists.

Failed to restore the registry (key= *Key_name*, filename= *File_name*). Possibly some applications might have keys open eg.,REGEDIT.EXE. Please close them.

The agent failed to restore the specified registry key.

Recommended Action: Close the application that accesses the specified registry keys. Contact Veritas Technical Support if the problem persists.

Duplicate key (or) Already part of notification (key=*Key_name*)

The specified registry key is already specified in the configuration.

Recommended Action: Verify that the values specified in Keys and ExcludeKeys attribute are unique.

Could not locate the file (file=*file_name*)

The agent was unable to locate the registry replication files.

Recommended Action: Verify that the files exist in the directory.

Failed to set *%s*

The agent failed to enable debug monitoring.

Recommended Action: Verify that monitor.exe is running. Contact Veritas Technical support if the problem persists.

Agent module and Registry-change-monitor module are not of the same version (received= *%s*, expected=*%s*)

The Registry Replication agent consists of two modules: Registry Replication and Registry Monitoring. If the versions of these modules are different, the agent will not come online.

Recommended Action: Verify the version of both agent modules. Contact Veritas Technical Support if problem persists.

Configuration Error : Key '*Key_name*' is configured in Replication Key List as well as in Exclude Key List

The specified key is configured both for Keys and ExcludeKeys attribute.

Recommended Action: Verify the configuration definition of the attributes Keys and ExcludeKeys. Make sure a specific key is not defined for both the attributes.

Configuration Error : Key '*Key_name*' configured in Replication Key List is a Sub Key of key '*Key_name*' configured in Exclude Key List

The specified key is configured for replication while its super key is configured for exclusion.

Recommended Action: Verify the configuration definition of the attributes Keys and ExcludeKeys.

Proxy agent error messages

Target Resource Name is not a valid name

The resource name specified in the TargetResName attribute is invalid.

Recommended Action: Verify the TargetResName attribute defined in the configuration file is assigned a correct value.

Target resource is not Probed

The agent cannot monitor the target resource because the target resource has not been probed.

Recommended Action: Verify that the target resource has been probed. To get the probed status of the target resource, execute `hares -display` from the command prompt.

Phantom agent error message

Contact Veritas Technical Support for any errors associated with the Phantom agent.

FileNone agent error message

File Name is not specified

The configuration file does not contain the path name, or the configured path name is incorrect.

Recommended Action: Verify that the attribute PathName is correctly defined in the configuration file.

ElifNone agent error message

FileName is not valid

The configuration file does not contain the path name, or the configured path name is incorrect.

Recommended Action: Verify that the attribute PathName is correctly defined in the configuration file.

FileOnOff agent error messages

Unable to create the file

The agent was unable to create the specified file.

Recommended Action: Verify that the filename specified in the PathName attribute does not contain any special characters, and the directory where the file will be created, exists on the node.

Unable to delete the file

The agent was unable to delete the specified file.

Recommended Action: Verify the filename specified in the PathName attribute exists, and the file is not locked by another user.

FileOnOnly agent error messages

ERROR! FileName is not valid

The configuration file does not contain the path name, or the configured path name is incorrect.

Recommended Action: Verify that the attribute PathName is correctly defined in the configuration file.

Unable to create the file

The agent was unable to create the specified file.

Recommended Action: Verify that the filename specified in the PathName attribute does not contain any special characters, and the directory where the file will be created, exists on the node.

Troubleshooting NetApp agents

This section describes the error messages for the VCS NetApp agents.

Failed to open connection to filer %s.

Recommended Action: Make sure that the VCS Helper Service account has domain admin privileges and is part of the administrator's group on the local host and the filer.

Make sure the private network is functioning properly. Verify you can ping the IP used for the private storage network. This is the IP defined the StorageIP attribute of the NetAppFiler resource.

Failed to initialize ONTAPI on system

The agent could not find the file NTAPADMIN.DLL on the system.

Recommended Action: Verify the file exists in the %VCS_HOME%\bin directory.

Invalid attributes exist in the configuration

Some agent attributes have not been defined or have been defined incorrectly.

Recommended Action: Verify the configuration definition for the agent.

ONTAP API called failed for object_name on filer_name.

The specified API failed on the specified object.

Recommended Action: See the NetApp ONTAP API documentation for information about the associated error message

Volume %s on filer %s is not a SnapMirror replicated volume

Recommended Action: Verify replication is set up on the specified volume.

Multiple snapmirror destinations for a volume is not supported by this agent. 'snapmirror status' for volume %s on filer %s returned multiple status entries. Administrative intervention required

Recommended Action: There should be only one destination per source volume.

Initialize VLibNetAppHost::Initialize() failed. (error_type: %s, error_code: 0x%s)

The agent could not detect the iSCSI or the FC Initiator on the host.

Recommended Action: Make sure that you have installed and configured Microsoft iSCSI Initiator or an FC Initiator on each node.

Failed to connect/disconnect virtual disk. (error_type: %s, error_code: 0x%s. error_message: %s)

This could occur because one or more of the following parameters are defined incorrectly in the VCS configuration:

- Filer name
- Volume name/LUN name
- Share name
- Storage IP

Recommended Action: Verify the configuration definition of the resource. Make sure each attribute is defined correctly.

Unable to create/delete online lock file %s. Error code %s,

Recommended Action: Make sure you have write permissions on the specified directory.