

Veritas™ Cluster Server Bundled Agents Reference Guide

ESX

5.1

Veritas Cluster Server Bundled Agents Reference Guide

Copyright © 2007 Symantec Corporation. All rights reserved.

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

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

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

The Licensed Software and Documentation are deemed to be “commercial computer software” and “commercial computer software documentation” as defined in FAR Sections 12.212 and DFARS Section 227.7202.

Symantec Corporation
20330 Stevens Creek Blvd.
Cupertino, CA 95014
www.symantec.com

Third-party legal notices

All third-party copyrights associated with this product are listed in the Third Party Copyrights document, which is included on the product disc.

Technical support

For technical assistance, visit <http://entsupport.symantec.com>.

Select phone or email support. Use the Knowledge Base search feature to access resources such as TechNotes, product alerts, software downloads, hardware compatibility lists, and our customer email notification service.

Contents

Chapter 1	Introduction	
	Resources and their attributes	11
	Modifying agents and their resources	12
	Attributes	12
Chapter 2	Storage agents	
	About the storage agents	15
	Disk agent	16
	Agent functions	16
	State definitions	16
	Attributes	16
	Resource type definition	16
Chapter 3	Network agents	
	About the network agents	17
	IP agent	18
	Dependency	18
	Agent functions	18
	State definitions	18
	Attributes	19
	Resource type definition	20
	Sample configurations	20
	Configuration 1	20
	Configuration using specified NetMask	20
	DNS agent	21
	Agent functions	21
	State definitions	21
	Attributes	22
	Resource type definition	23
	Online queries	23
	Setting the Alias attribute	24
	Setting the IPAddress attribute	24
	Monitor scenarios	25
	Sample web server configuration	25
	Sample DNS configuration	26

DNS agent prerequisites	26
Secure DNS update	26
Setting up secure updates using TSIG keys	26

Chapter 4 Virtualization management agents

About the virtualization management agents	29
ESXVirtualMachine agent	30
Dependencies	30
Requirements	30
Agent functions	30
State definitions	31
Attributes	31
Resource type definition	34
Verifying if a service group can be migrated	34
Sample configurations	35
Basic Suse configuration	35
ESXVirtualMachine agent configured for migration	35
GuestOSApp agent	36
Dependencies	36
Requirements	36
Agent functions	36
State definitions	36
Attributes	37
Resource type definition	37
VMFSVolume agent	38
Agent functions	38
State definitions	38
Attributes	38
Resource type definition	38
Sample configuration	38
VMIP agent	39
Requirements	39
Agent functions	39
State definitions	39
Attributes	40
Resource type definition	41
Sample configuration	41
VSwitch agent	42
Agent functions	42
State definitions	42
Attribute	42
Resource type definition	43
Sample configuration	43

	ESXHost agent	44
	Prerequisites	44
	Agent functions	44
	Attribute	44
	Resource type definition	45
Chapter 5	Service and application agents	
	About the service and application agents	47
	Application agent	48
	Dependencies	48
	Agent functions	48
	State definitions	49
	Attributes	50
	Resource type definition	52
	Sample configurations	53
	Configuration 1	53
	Configuration 2	53
	Process agent	54
	Dependencies	54
	Agent functions	54
	State definitions	54
	Attributes	55
	Resource type definition	56
	Sample configurations	56
	Configuration	56
	ProcessOnOnly agent	57
	Agent functions	57
	State definitions	57
	Attributes	57
	Resource type definition	59
	Sample configurations	59
	Configuration 1	59
	Configuration 2	59
Chapter 6	Infrastructure and support agents	
	About the infrastructure and support agents	61
	NotifierMngr agent	62
	Dependency	62
	Agent functions	62
	State definitions	62
	Attributes	63
	Resource type definition	66

Sample configuration	67
Configuration	67
VRTSWebApp agent	69
Agent functions	69
State definitions	69
Attributes	69
Resource type definition	70
Sample configuration	70
Proxy agent	71
Agent functions	71
Attributes	71
Resource type definition	72
Sample configurations	72
Configuration 1	72
Configuration 2	72
Configuration 3	72
Phantom agent	74
Agent functions	74
Resource type definition	74
Sample configurations	74
Configuration 1	74
Configuration 2	74

Chapter 7

Testing agents

About the program support agents	77
ElifNone agent	78
Agent function	78
Attributes	78
Resource type definition	78
Sample configuration	78
FileNone agent	79
Agent functions	79
Attribute	79
Resource type definition	79
Sample configuration	79
FileOnOff agent	80
Agent functions	80
Attribute	80
Resource type definition	80
Sample configuration	80

FileOnOnly agent	81
Agent functions	81
Attribute	81
Resource type definition	81
Sample configuration	81
Glossary	83
Index	85

Introduction

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

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

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

Agents can:

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

For a more detailed overview of agents, see the VCS User's Guide.

Resources and their attributes

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

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

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

Modifying agents and their resources

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

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

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

Attributes

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

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

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

Table 1-1 Attribute data types

Data Type	Description
string	<p>Enclose strings, which are a sequence of characters, in double quotes (""). Optionally enclose strings in quotes when they begin with a letter, and contains only letters, numbers, dashes (-), and underscores (_).</p> <p>A string can contain double quotes, but the quotes must be immediately preceded by a backslash. In a string, represent a backslash with two slashes (//).</p>
integer	<p>Signed integer constants are a sequence of digits from 0 to 9. You can precede them with a dash. They are base 10. Integers cannot exceed the value of a 32-bit signed integer: 21471183247.</p>

Table 1-1 Attribute data types

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

Table 1-2 Attribute dimensions

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

Storage agents

This chapter contains:

- [“Disk agent”](#) on page 16

About the storage agents

Use the storage agents to monitor shared storage.

Disk agent

Monitors a disk.

Agent functions

Monitor Determines if the disk is accessible by performing read I/O operations on the raw disk.

State definitions

FAULTED Indicates that the disk has unexpectedly stopped working.

UNKNOWN Indicates that a problem exists either with the configuration or the ability to determine the status of the resource.

Attributes

Table 2-1 Required attributes

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

Resource type definition

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

Network agents

This chapter contains:

- [“IP agent”](#) on page 18
- [“DNS agent”](#) on page 21

About the network agents

Use network agents to provide high availability for networking resources.

IP agent

Manages the process of configuring a virtual IP address and its subnet mask on an interface. The interface must be enabled with a physical (or administrative) base IP address before you can assign it a virtual IP address. The virtual IP address must not be in use.

Dependency

IP resources depend on NIC resources (except for VCS for ESX).

Agent functions

Online	Configures the IP address to the NIC. Checks if another system is using the IP address. Uses the <code>ifconfig</code> command to set the IP address on a unique alias on the interface. Linux and ESX: Sends out a gratuitous ARP. Linux and ESX: Sends out a gratuitous ARP.
Offline	Brings down the IP address specified in the Address attribute.
Monitor	Monitors the interface to test if the IP address that is associated with the interface is alive.
Clean	Brings down the IP address associated with the specified interface.

State definitions

ONLINE	Indicates that the device is up and the specified IP address is assigned to the device.
OFFLINE	Indicates that the device is down or the specified IP address is not assigned to the device.
UNKNOWN	Indicates that the agent could not determine the state of the resource or that the resource attributes are invalid.

Attributes

Table 3-1 Required attributes

Required attribute	Description
Address	<p>A virtual IP address, different from the base IP address, which is associated with the interface—the address you specify must not be the same as the configured physical IP address.</p> <p>Type and dimension: string-scalar</p> <p>Example: "192.203.47.61"</p>
Device	<p>The name of the NIC device associated with the IP address. Requires the device name without an alias.</p> <p>Type and dimension: string-scalar</p> <p>Example: Specify eth0 to assign the IP address to the next available alias. Use the <code>ifconfig -a</code> command to display a list of NICs that are up and the IP addresses assigned to each NIC.</p>

Table 3-2 Optional attributes

Optional attribute	Description
NetMask	<p>Subnet mask associated with the IP address. Specify the value of NetMask in decimal (base 10).</p> <p>If Netmask is not specified, the agent uses the operating system's default netmask.</p> <p>Type and dimension: string-scalar</p> <p>Example: "255.255.255.0"</p>
Options	<p>Options for the <code>ifconfig</code> command.</p> <p>Type and dimension: string-scalar</p> <p>Example: "broadcast 172.20.9.255"</p>

Resource type definition

```
type IP (  
    static keylist SupportedActions = { "device.vfd", "route.vfd" }  
    static str ArgList[] = { Device, Address, NetMask, Options }  
    str Device  
    str Address  
    str NetMask  
    str Options  
)
```

Sample configurations

Configuration 1

```
IP          IP_192_203_47_61 (  
    Device = eth0  
    Address = "192.203.47.61"  
)
```

Configuration using specified NetMask

```
IP          IP_192_203_47_61 (  
    Device = eth0  
    Address = "192.203.47.61"  
    NetMask = "255.255.248.0"  
)
```

DNS agent

The DNS agent updates and monitors the host name to IP (A record) and the canonical name (CNAME) mapping in the domain name server when failing over virtual machines across subnets. Failing over virtual machines across subnets is also called a wide-area failover.

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

Agent functions

Online	Queries the authoritative name server of the domain for the A record or the CNAME record and updates the A record or the CNAME record on the name server with the specified alias to canonical name mapping and hostname to IP address mapping. Adds a new A record or CNAME record if a related record is not found. Creates an Online lock file if the Online function was successful.
Offline	Removes the Online lock file, which the Online agent function created.
Monitor	If the Online lock file exists, the Monitor function queries the name servers for the A record or the CNAME record for the alias. It reports back online if the response from at least one of the name servers contains the same canonical name associated with the alias in the Hostname attribute. If no servers return the appropriate name, the monitor reports the resource as offline.
Clean	Removes the Online lock file, if it exists.
Open	Removes the Online lock file if the Online lock file exists, and the A record or the CNAME record on the name server does not contain the expected alias or canonical name mapping hostname to IP address.

State definitions

ONLINE	An Online lock exists and the A record or the CNAME RR is as expected.
OFFLINE	Either the Online lock does not exist, or the expected record is not found.
UNKNOWN	A problem exists with the configuration.

Attributes

Table 3-3 Required attributes

Required attribute	Description
Domain	A string representing the domain name. Type and dimension: string-scalar Example: "veritas.com"
Hostname	A string representing canonical name of a system. Type and dimension: string-scalar Example: "mtv.veritas.com"
TTL	A non-zero integer representing the Time To Live (TTL) value, in seconds, for the DNS entries in the zone you are updating. A lower value means more hits on your DNS server, while a higher value means more time for your clients to learn about changes. Type and dimension: integer-scalar Default: 86400 Example: "3600"

Table 3-4 Optional attributes

Optional attribute	Description
Alias	You must set either the Alias or the IPAddress attribute or both. A string representing the alias to the canonical name. Type and dimension: string-scalar Example: "www" Where www is the alias to the canonical name mtv.veritas.com.

Table 3-4 Optional attributes

Optional attribute	Description
IPAddress	You must set either the IPAddress or the Alias attribute or both. Specifies the IP address assigned to the hostname. Type and dimension: string-scalar Example: "10.123.12.55"
StealthMasters	The list of primary master name servers in the domain. Optional if the zone's name server record lists the primary master name server. If the primary master name server is a stealth server, define this attribute. A stealth server is a name server that is authoritative for a zone, but is not listed in that zone's name server records. Type and dimension: string-keylist Example: { "10.190.112.23" }
TSIGKeyFile	Required when you configure DNS for secure updates. Specifies the absolute path to the file containing the private TSIG (Transaction Signature) key. Type and dimension: string-scalar Example: "/var/tsig/Kveritas.com.+157+00000.private"

Resource type definition

```
type DNS (  
    static str ArgList[] = { Domain, Alias, Hostname, IPAddress,  
        TTL, TSIGKeyFile, StealthMasters }  
    str Domain  
    str Alias  
    str Hostname  
    str IPAddress  
    int TTL = 86400  
    str TSIGKeyFile  
    str StealthMasters[]  
)
```

Online queries

The Online function performs different kinds of queries depending on whether you specify the Alias or IPAddress attributes, or if you specify both.

If you specify both the Alias and IPAddress attributes both of the following sections are true:

- [“Setting the Alias attribute”](#) on page 24
- [“Setting the IPAddress attribute”](#) on page 24

Setting the Alias attribute

If the canonical name in the response CNAME record does not match the one specified for the resource, the Online function tries to update the CNAME record on all authoritative master name servers in its domain. The master name servers that it can reach and where it has update permission. If the DNS update was successful, or was not necessary on at least one of the name servers, the Online function creates an Online lock file. The monitor agent function checks for the existence of this file. The Online agent function does not create the Online lock file if it is unable to update at least one domain name server.

Setting the IPAddress attribute

If the IP address in the response A record does not match the one specified for the resource, the Online function tries to update the A record on all authoritative master name servers in its domain (those master name servers that it can reach and where it has update permission). If the DNS update was successful, or was not necessary on at least one of the name servers, the Online function creates an Online lock file. The monitor agent function checks for the existence of this file. The Online agent function does not create the Online lock file if it is unable to update at least one domain name server.

A stealth server is a name server that is authoritative for a zone, but is not listed in that zone’s NS records. If you specify the StealthMasters attribute, the Online agent function tries to update the name servers specified in the StealthMasters attribute.

In BIND 8 and above, the primary master name server on receiving an update sends notification (NOTIFY) to all its slave servers asking them to pick up the update.

Monitor scenarios

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

Table 3-5 Monitor scenarios for the Online lock file

Online lock file exists	Expected CNAME RR or A record	Monitor returns
NO	N/A	OFFLINE
YES	NO	OFFLINE
YES	YES	ONLINE

Note: The DNS agent supports BIND version 8 and above.

Sample web server configuration

Take the former Veritas corporate web server as an example. A person using a web browser specifies the URL `www.veritas.com` to view the Veritas web page, where `www.veritas.com` maps to the canonical name `mtv.veritas.com`, which is a host in Mountain View running the web server. The browser, in turn, retrieves the IP address for the web server by querying the domain name servers. If the web server for `www.veritas.com` is failed over from Mountain View to Heathrow, the domain name servers need to be updated with the new canonical name mapping so that the web browsers are directed to Heathrow instead of Mountain View. In this case, the DNS agent should update the name server to change the mapping of `www.veritas.com`, from `mtv.veritas.com` to the canonical name of the standby system in Heathrow, `hro.veritas.com`, in case of a failover.

Sample DNS configuration

```
DNS www (  
    Domain = "example.com"  
    Alias = www  
    Hostname = virtual1  
)
```

Bringing the `www` resource online updates the authoritative nameservers for domain `example.com` with the following CNAME record:

```
■ ESX  
    www CNAME wwwvirtual1
```

All DNS lookups for `www.example.com` resolve to `www.virtual1.example.com`.

DNS agent prerequisites

For the DNS agent to work correctly, set up the primary and the stealth masters to accept and correctly process updates. You can test this via the `nsupdate` command that ships with the operating system—refer to the `nsupdate` manpage for usage. By default both SuSE and Red Hat install with SELinux enabled and configured to disallow the DNS server process named, to modify the on-file DNS database—the zone files. You must set up SELinux on the DNS server systems to allow the `named` process to update. Refer to the operating system documentation for more information.

Secure DNS update

The DNS agent by default—when the attribute `TSIGKeyFile` is unspecified—expects the IP address of the hosts that can update the DNS records dynamically to be specified in the `allow-updates` field of the zone. However, since IP addresses can be easily spoofed, a secure alternative is to use TSIG (Transaction Signature) as specified in RFC 2845. TSIG is a shared key message authentication mechanism available in DNS. A TSIG key provides a means to authenticate and verify the validity of DNS data exchanged, using a shared secret key between a resolver and either one or two servers.

Setting up secure updates using TSIG keys

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

To use secure updates using TSIG keys

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

```
# dnssec-keygen -a HMAC-MD5 -b 512 -n HOST veritas.com.  
Kveritas.com.+157+00000
```

- 2 Open the `Kveritas.com.+157+00000.key` file. After running the `cat` command, the contents of the file resembles:

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

- 3 Copy the shared secret (the TSIG key), which looks like:
`+Cdjlkef9ZTSeixERZ433Q==`
- 4 Configure the DNS server to only allow TSIG updates using the generated key. Open the `named.conf` file and add these lines.

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

Where `+Cdjlkef9ZTSeixERZ433Q==` is the key.

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

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

- 6 Save and restart the `named` process.
- 7 Place the files containing the keys on each of the nodes that is listed in your group's `SystemList`. The DNS agent uses this key to update the name server. Copy both the private and public key files on to the node. A good location is in the `/var/tsig/` directory.
- 8 Set the `TSIGKeyFile` attribute for the DNS resource to specify the file containing the private key.

```
DNS www (
  Domain = "veritas.com"
  Alias = www
  Hostname = north
  TSIGKeyFile a= "/var/tsig/Kveritas.com.+157+00000.private"
)
```


Virtualization management agents

This chapter contains:

- [“ESXVirtualMachine agent”](#) on page 30
- [“GuestOSApp agent”](#) on page 36
- [“VMFSVolume agent”](#) on page 38
- [“VMIP agent”](#) on page 39
- [“VSwitch agent”](#) on page 42
- [“ESXHost agent”](#) on page 44

About the virtualization management agents

Virtualization management agents allow you to manage certain virtualized environments and maintain their high availability.

ESXVirtualMachine agent

Brings online, takes offline, and monitors virtual machines that are configured on the ESX Server.

Dependencies

This resource can depend on the VMFSVolume and the VSwitch resources for its datastore and network.

Requirements

This agent requires VMware Tools to operate. You must also make sure that the guestinfo interfaces of the VMware Virtual Machine tools are enabled. Note that these interfaces are enabled by default.

Agent functions

Online	Registers the virtual machine if it is not already registered. Starts the virtual machine. The Online agent function fails with an error message if the ESX Server host is in maintenance mode.
Offline	Attempts a graceful shut down of the virtual machine.
Monitor	Detects the virtual machine's state.
Action	<ul style="list-style-type: none">■ testVCCconnect This agent function verifies that you can connect to the VirtualCenter server with the current values of the attributes for this resource. See “Verifying if a service group can be migrated” on page 34.■ migrate This agent function helps to migrate the virtual machine from one node to another. This action script is for internal use only, and Symantec recommends that you do not use it. For migrating virtual machines, use the <code>hagrps -migrate</code> command. For more information, see the <i>Veritas Cluster Server for VMware ESX Implementation Guide</i>.
Clean	Forcefully shuts down the virtual machine.
Close	Cleans up internal agent datastructures.

State definitions

ONLINE	Indicates that the virtual machine is up and has a heartbeat.
OFFLINE	Indicates that the virtual machine is up but is not sending a heartbeat, or is down but still registered on the ESX Server.
FAULTED	Indicates that the virtual machine is up but not sending a heartbeat, or is down but still registered on the ESX Server.
UNKNOWN	Indicates the agent cannot determine the virtual machine's state.

Attributes

Table 4-1 Required attribute

Required attribute	Description
CfgFile	<p>Specifies the complete pathname of the configuration file for a virtual machine. The pathname starts with the slash (/) preceding the file name. Do not use the alias. Use the UUID path.</p> <p>When you configure a virtual machine for high availability using the Veritas Virtualization Manager, the value of this attribute is set automatically.</p> <p>Type and dimension: string-scalar</p> <p>Example: /vmfs/volumes/5c5d8e06-da11f7ce-7892-930b00a53cd2/sles9v2/sles9v2.vmx</p>

Table 4-2 Optional attributes

Optional attribute	Description
MonitorHB	<p>This flag enables heartbeat monitoring for a virtual machine. When the value of this attribute is 1, the agent detects a heartbeat failure of the virtual machine, and flags it as a fault to VCS.</p> <p>The virtual machine must have VMware Tools installed and running inside of the virtual machine.</p> <p>Type and dimension: boolean-scalar</p> <p>Default: "1"</p>
username	<p>The username to connect to the VirtualCenter Server.</p> <p>Migrate functionality requires that you set this attribute.</p> <p>When you configure a virtual machine for high availability using the Veritas Virtualization Manager, the value of this attribute is set automatically.</p> <p>Type and dimension: string-scalar</p>
password	<p>The password to connect to the VirtualCenter Server. This password must be encrypted. For encryption, use the vscencrypt utility.</p> <p>For more information on using the vscencrypt utility, see the <i>Veritas Cluster Server User's Guide</i>.</p> <p>Migrate functionality requires that you set this attribute.</p> <p>When you configure a virtual machine for high availability using the Veritas Virtualization Manager, the value of this attribute is set automatically.</p> <p>Type and dimension: string-scalar</p>

Table 4-2 Optional attributes

Optional attribute	Description
sslcert	<p>The path to the keystore file for the VirtualCenter Server.</p> <p>Migrate functionality requires that you set this attribute.</p> <p>You must copy the keystore file to each node in the cluster. Use the same path on each node to save time. Set the value of the sslcert attribute to the full pathname of the keystore file on each VCS node.</p> <p>The value of this attribute is not set automatically by the Veritas Virtualization Manager.</p> <p>For more information on creating and using keystore files, see the <i>Veritas Cluster Server for VMware ESX Implementation Guide</i>.</p> <p>Type and dimension: string-scalar</p>
esxhostdomain	<p>The domain name of the ESX host, from the VirtualCenter Server's perspective. If you are unsure, check the ESX Server names when connected to the VirtualCenter Server through your preferred client.</p> <p>Migrate functionality requires that you set this attribute.</p> <p>When you configure a virtual machine for high availability using the Veritas Virtualization Manager, the value of this attribute is set automatically.</p> <p>Type and dimension: string-scalar</p>
vmname	<p>The display name of this virtual machine. If you have set your resource name to be the same as the display name, skip this attribute.</p> <p>Migrate functionality requires that you set this attribute.</p> <p>When you configure a virtual machine for high availability using the Veritas Virtualization Manager, the value of this attribute is set automatically.</p> <p>Type and dimension: string-scalar</p>

Table 4-2 Optional attributes

Optional attribute	Description
VCserver	<p>The name or the IP address of the Virtual Infrastructure Server.</p> <p>When you configure a virtual machine for high availability using the Veritas Virtualization Manager, the value of this attribute is set automatically.</p> <p>Type and dimension: string-scalar</p>

Resource type definition

```

type ESXVirtualMachine (
  static int IntentionalOffline = 1
  static keylist SupportedActions = { growfs, migrate }
  static boolean Migratable = 1
  static keylist ExternalStateChange = { OnlineGroup,
  OfflineGroup }
  static str ArgList[] = { CfgFile, MonitorHB, VCserver, username,
  password, sslcert, esxhostdomain, vmname }
  str CfgFile
  boolean MonitorHB = 1
  str VCserver
  str username
  str password
  str sslcert
  str esxhostdomain
  str vmname
)

```

Verifying if a service group can be migrated

You can use the testVCCconnect agent function to verify that the ESXVirtualMachine resource attributes are correctly configured for connecting to the VirtualCenter server.

Run the testVCCconnect script on each node of the cluster, if you plan to use hagr -migrate functionality. You can find the script in the actions directory /opt/VRTSvcs/bin/ESXVirtualMachine/actions/

Run the testVCCconnect function:

```
# hares -action resname token -sys vcssystemname
```

The following line is an example:

```
# hares -action evm testVCCconnect -sys esxNode1
```

Where `evm` is the name of the `ESXVirtualMachine` resource, `testVCCConnect` is the token name, and `esxNode1` is the name of the node where you want to test the connection from.

For more information on `hagr` -migrate functionality, see the:

- *VCS for VMware ESX Implementation Guide*

Sample configurations

Basic Suse configuration

```
ESXVirtualMachine sles9vm3 (  
  CfgFile = "/vmfs/volumes/44a15b88-f73de898-a4c3-00093d10858b/  
  vm3/vm3.vmx"  
)
```

ESXVirtualMachine agent configured for migration

```
ESXVirtualMachine vm3 (  
  CfgFile = "/vmfs/volumes/44a15b88-f73de898-a4c3-00093d10858b/  
  vm3/vm3.vmx"  
  VCserver = pc54  
  username = Administrator  
  password = dllLe  
  sslcert = "/root/vmware-certs/pc54.keystore"  
  esxhostdomain = "veritas.com"  
  vmname = vm3  
)
```

GuestOSApp agent

Reflects the state of an application that runs in a guest operating system.

Dependencies

This resource depends on the ESXVirtualMachine resource.

Requirements

This agent requires VMware Tools to operate. You must also make sure that the `guestinfo` interfaces of the VMware Virtual Machine tools are enabled.

Agent functions

Action	<p>The action function tests if the GuestOSApp resource is properly configured. The Action functions that are implemented can return:</p> <ul style="list-style-type: none">■ <code>fault</code> VCS uses this agent function internally.■ <code>getappstate</code> This agent function returns the state of the application that runs inside a virtual machine as set in the VMWare <code>guestinfo</code> interface. Typically, you can use this action to verify if a GuestOSApp resource (and a corresponding Application resource inside the virtual machine) are properly configured and mapped. The action script takes no parameters and can be run using your preferred VCS management interface. If you use a command line interface, run the command: <pre>hares -action resname getappstate -sys vcssystemname</pre>
Monitor	<p>Probes the state of the application through the VMware ESX virtual machine <code>guestinfo</code> variable and reflects its state accordingly.</p>

State definitions

ONLINE	<p>If the VMware ESX virtual machine <code>guestinfo</code> variable state suggests that the application is <code>ONLINE</code>, the agent returns an <code>ONLINE</code> state.</p>
FAULTED	<p>If the VMware ESX virtual machine <code>guestinfo</code> variable state suggests that the application is <code>OFFLINE</code>, the agent returns an <code>FAULTED</code> state.</p>

Attributes

Table 4-3 Required attribute

Required attribute	Description
VMwareResName	Use the exact name as it appears for the ESXVirtualMachine resource in the VCS configuration. Type and dimension: string-scalar

Resource type definition

```
type GuestOSApp (  
  static int IntentionalOffline = 1  
  static int MonitorInterval = 60  
  static int OnlineWaitLimit = 5  
  static keylist SupportedActions = { "fault"  
  static str ArgList[] = { "VMwareResName:CfgFile" }  
  str VMwareResName  
)
```

VMFSVolume agent

Monitors the VMFS volumes on an ESX Server

Agent functions

Monitor Checks for the specified volume. If it exists, the agent reports as online. If it does not exist, the resource faults.

State definitions

ONLINE Defines a mounted and accessible data store on the ESX host.

Attributes

Table 4-4 Required attribute

Required attribute	Description
Volume	Specifies the volume-UID of the VMFS datastore that you want to monitor. The agent returns online if the datastore is mounted and accessible on the ESX host. Type and dimension: string-scalar Example: /vmfs/volumes/5c5d8e06-da11f7ce-7892-930b00a53cd2

Resource type definition

```
type VMFSVolume (  
    static int MonitorInterval = 30  
    static str ArgList[] = { Volume }  
    str Volume[]  
)
```

Sample configuration

```
VMFSVolume Shared3 (  
    Volume = "/vmfs/volumes/5c5d8e06-da11f7ce-7892-930b00a53cd2"  
)
```

VMIP agent

The virtual machine IP (VMIP) agent updates the IP address, subnet mask, gateway, and DNS on an interface in a virtual machine.

Requirements

This agent requires VMware Tools to operate. You must also make sure that the guestinfo interfaces of the VMware Virtual Machine tools are enabled.

Agent functions

Online	Configures the IP address, subnet mask, gateway, and the DNS to the virtual NIC. Creates an Online lock file if the Online function was successful.
Offline	Removes the Online lock file, which the Online agent function creates.
Monitor	Checks if the Online lock file exists. Reports back in an ONLINE state if the file exists.
Clean	Removes the Online lock file.
Open	Removes the Online lock file.

State definitions

ONLINE	Indicates that the device is up and the specified IP address is assigned to the device.
OFFLINE	Indicates that the device is down or the specified IP address is not assigned to the device.
UNKNOWN	Indicates that the agent could not determine the state of the resource or that the resource attributes are invalid.

Attributes

Table 4-5 Required attribute

Required attribute	Description
VMwareResName	<p>The name of the VCS resource that manages the virtual machine.</p> <p>Type and dimension: string-scalar</p>
IPAddress	<p>The IP address that is assigned to the virtual machine interface.</p> <p>Type and dimension: string-scalar</p> <p>Example: "10.123.12.55"</p>
MACAddress	<p>The MAC address of the virtual NIC, which must be the MAC address that the ESX Server assigns to the virtual NIC that is inside the virtual machine.</p> <p>Type and dimension: string-scalar</p> <p>Example: "00:0C:29:FF:86:2E"</p>
NetMask	<p>The subnet mask that is associated with the IP address. You must specify this value in decimal (base 10).</p> <p>Type and dimension: string-scalar</p> <p>Example "255.255.255.0"</p>
Gateway	<p>The default gateway for the virtual machine.</p> <p>Type and dimension: string-scalar</p> <p>Example: "10.123.17.1"</p>
DNS	<p>List of DNS servers in the required search order.</p> <p>Type and dimension: string-keylist</p> <p>Example: {"10.123.17.160", "10.123.17.161"}</p>

Resource type definition

```
type VMIP (  
    static int MonitorInterval = 300  
    static str ArgList[] = { "VMwareResName:CfgFile", IPAddress,  
        MACAddress, NetMask, Gateway, DNS }  
    str VMwareResName  
    str IPAddress  
    str MACAddress  
    str NetMask  
    str Gateway  
    str DNS[]  
)
```

Sample configuration

```
VMIP vmip (  
    VMwareResName = vm  
    IPAddress = "10.123.12.55"  
    MACAddress = "00:0C:29:FF:86:2E"  
    NetMask = "255.255.2255.0"  
    Gateway = "10.123.17.1"  
    DNS = { "10.123.17.160", "10.123.17.161" }  
)
```

VSwitch agent

Monitors a virtual switch on an ESX Server.

Agent functions

Monitor The agent performs a hardware link test using the mii-tool, and returns online if any of the switch's uplinks are up.

State definitions

ONLINE Indicates that the virtual switch is working.

FAULTED Indicates that the virtual switch has failed.

UNKNOWN Indicates the agent cannot determine the virtual switch's interface states. This state may be due to an incorrect configuration.

Attribute

Table 4-6 Required attribute

Required attribute	Description
VirtualSwitch	<p>Specifies the name of the virtual switch that you want to monitor.</p> <p>An ESXVirtualMachine resource that depends on a VSwitch resource, requires that all the VSwitch resource names be identical across all the cluster nodes. Note that VCS automatically makes sure that the names are identical when you use the Veritas Virtualization Manager.</p> <p>Type and dimension: string-scalar</p> <p>Example: "vSwitch0"</p>

Resource type definition

```
type VSwitch (  
  static int MonitorInterval = 30  
  static str ArgList[] = { VirtualSwitch }  
  static str Operations = None  
  str VirtualSwitch  
)
```

Sample configuration

```
VSwitch vSwitch1 (  
  VirtualSwitch = vSwitch0  
)
```

ESXHost agent

The ESXHost agent, an internal-use only agent, enforces compatibility between VCS and VMware’s DRS and maintenance modes.

The ESXHost agent is automatically deployed on each node in the cluster, and is created in a service group called the ESXHostServiceGroup. This is a standalone, parallel service group, which you should not modify. The ESXHost agent, and its corresponding service group, are required on each node of your VCS for VMware ESX cluster.

The ESXHost agent enforces a two-way compatibility between VCS and VMware’s DRS and maintenance mode functions. If you bring down a node with VMware maintenance mode to perform system maintenance tasks, the ESXHost agent evacuates all VCS service groups to another ESX node in the VCS cluster and performs a system freeze for the evacuated node. The agent also provides reverse compatibility—if it detects that a VCS system is frozen, and no active virtual machines are on the host, it brings down the host to VMware maintenance mode.

Prerequisites

For the ESXHost agent to work properly, ensure that an exact match exists between the overlapping nodes on VCS and VMware clusters. Make sure that the DRS cluster uses exactly the same hosts as the hosts that overlap with the VCS cluster.

Agent functions

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

Attribute

Table 4-7 Internal attributes

Required attribute	Description
OfflineMonitorInterval	For internal use only. Do not modify.

Resource type definition

```
type ESXHost (  
    static int OfflineMonitorInterval = 60  
    static str Operations = None  
)
```


Service and application agents

This chapter contains:

- [“Application agent”](#) on page 48
- [“Process agent”](#) on page 54
- [“ProcessOnOnly agent”](#) on page 57

About the service and application agents

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

Application agent

Brings applications online, takes them offline, and monitors their status. Enables you to specify different executables for the online, offline, and monitor routines, because most applications have executables to start and stop the application. The executables must exist locally on each node.

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

The agent starts and stops the application with user-specified programs.

You can monitor the application in the following ways:

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

This agent supports the AMF feature. For more information about the AMF feature, refer to the Veritas Cluster Server User's Guide.

Dependencies

Depending on how you plan to use it, this type of resource can depend on IP, IPMultiNIC, and Mount resources.

Agent functions

Online	Runs the StartProgram with the specified parameters in the context of the specified user.
Offline	Runs the StopProgram with the specified parameters in the context of the specified user.
Monitor	<p>If you specify the MonitorProgram, the agent executes the user-defined MonitorProgram in the user-specified context. If you specify PidFiles, the routine verifies that the process ID found in each listed file is running. If you specify MonitorProcesses, the routine verifies that each listed process is running in the context you specify.</p> <p>Use any combination among these attributes (MonitorProgram, PidFiles, or MonitorProcesses) to monitor the application.</p> <p>If any one process specified in either PidFiles or MonitorProcesses is determined not to be running, the monitor returns OFFLINE. If the process terminates ungracefully, the monitor returns OFFLINE and failover occurs.</p>

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

State definitions

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

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

UNKNOWN Indicates an indeterminable application state or invalid configuration.

Attributes

Table 5-1 Required attributes

Required attribute	Description
StartProgram	<p>The executable, created locally on each node, which starts the application. Specify the complete path of the executable. Applicable command line arguments follow the name of the executable and are separated by spaces.</p> <p>Note: Do not use the opening and closing ({}) brace symbols in this string.</p> <p>Type and dimension: string-scalar</p> <p>Example: "/usr/sbin/samba start"</p>
StopProgram	<p>The executable, created locally on each node, that stops the application. Specify the complete path of the executable. Applicable command line arguments follow the name of the executable and are separated by spaces.</p> <p>Note: Do not use the opening and closing ({}) brace symbols in this string.</p> <p>Type and dimension: string-scalar</p> <p>Example: "/usr/sbin/sample_app stop"</p>
<p>At least one of the following attributes:</p> <ul style="list-style-type: none"> ■ MonitorProcesses ■ MonitorProgram ■ PidFiles 	<p>See "Optional attributes" on page 51.</p>

Table 5-2 Optional attributes

Optional attribute	Description
CleanProgram	<p>The executable, created locally on each node, which forcibly stops the application. Specify the complete path of the executable. Applicable command line arguments follow the name of the executable and are separated by spaces.</p> <p>Type and dimension: string-scalar</p>
MonitorProcesses	<p>A list of processes that you want monitored and cleaned. Each process name is the name of an executable. Qualify the executable name with its complete path if the path starts the executable.</p> <p>The process name must be the name displayed by the <code>ps -ef</code> command for the process.</p> <p>Type and dimension: string-vector</p> <p>Example: "nmbd"</p>
MonitorProgram	<p>The executable, created locally on each node, which monitors the application. Specify the complete path of the executable. Applicable command line arguments follow the name of the executable and are separated by spaces.</p> <p>MonitorProgram can return the following VCSAgResState values: OFFLINE value is 100; ONLINE values range from 101 to 110 (depending on the confidence level); 110 equals confidence level of 100%. Any other value = UNKNOWN.</p> <p>Note: Do not use the opening and closing ({}) brace symbols in this string.</p> <p>Type and dimension: string-scalar</p>

Table 5-2 Optional attributes

Optional attribute	Description
PidFiles	<p>A list of PID files that contain the process ID (PID) of the processes that you want monitored and cleaned. These are application generated files. Each PID file contains one monitored PID. Specify the complete path of each PID file in the list.</p> <p>The process ID can change when the process restarts. If the application takes time to update the PID file, the agent's monitor script may return an incorrect result. If this occurs, increase the ToleranceLimit in the resource definition.</p> <p>Type and dimension: string-vector</p>
User	<p>The user ID for running StartProgram, StopProgram, MonitorProgram, and CleanProgram. The processes specified in the MonitorProcesses list must run in the context of the specified user. Monitor checks the processes to make sure they run in this context.</p> <p>Type and dimension: string-scalar</p> <p>Default: root</p>

Resource type definition

```

type Application (
  static keylist SupportedActions = { "program.vfd", "user.vfd",
  "cksum.vfd", getcksum }
  static str ArgList[] = { User, StartProgram, StopProgram,
  CleanProgram, MonitorProgram, PidFiles, MonitorProcesses }
  str User
  str StartProgram
  str StopProgram
  str CleanProgram
  str MonitorProgram
  str PidFiles[]
  str MonitorProcesses[]
)

```

Sample configurations

Configuration 1

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

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

Configuration 2

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

```
Application samba_app2 (  
    StartProgram = "/usr/sbin/samba start"  
    StopProgram = "/usr/sbin/samba stop"  
    CleanProgram = "/usr/sbin/samba force stop"  
    MonitorProgram = "/usr/local/bin/sambaMonitor all"  
    MonitorProcesses = { "smbd", "nmbd" }  
)
```

Process agent

Starts, stops, and monitors a process that you specify.

This agent supports the AMF feature. For more information about the AMF feature, refer to the Veritas Cluster Server User's Guide.

Dependencies

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

Agent functions

Online	Starts a process in the background with optional arguments and priority in the specified user context.
Offline	Terminates the process with a <code>SIGTERM</code> . If the process does not exit, a <code>SIGKILL</code> is sent.
Monitor	Checks to see if the process is running by scanning the process table for the name of the executable pathname and argument list.
Clean	Terminates all ongoing resource actions and takes the resource offline, forcibly when necessary.

State definitions

ONLINE	Indicates that the specified process is running in the specified user context.
OFFLINE	Indicates that the specified process is not running in the specified user context.
FAULTED	Indicates that the process has terminated unexpectedly.
UNKNOWN	Indicates that the agent can not determine the state of the process.

Attributes

Table 5-3 Required attribute

Required attribute	Description
PathName	<p>Complete pathname to access an executable program. This path includes the program name. If a script controls the process, the PathName defines the complete path to the shell.</p> <p>This attribute must not exceed 256 characters.</p> <p>Type and dimension: string-scalar</p> <p>Example: <code>"/usr/sbin/proc1"</code></p>

Table 5-4 Optional attributes

Optional attribute	Description
Arguments	<p>Passes arguments to the process. If a script controls the process, the script is passed as an argument. Separate multiple arguments with a single space. A string cannot accommodate more than one space between arguments, nor allow for leading or trailing whitespace characters.</p> <p>Type and dimension: string-scalar</p>
PidFile	<p>The file that contains the process ID for the monitoring process. Specify the PidFile attribute for the monitoring process to use the Pid. Otherwise, to complete the monitoring process the agent uses the ps output.</p> <p>Note that when you use scripts, or other indirect mechanisms, to start processes, you must set the PidFile attribute if the ps output is different from the configured values for the PathName or Arguments attributes.</p> <p>Type and dimension: string-scalar</p> <p>Example: <code>"/var/lock/sendmail.pid"</code></p>

Table 5-4 Optional attributes

Optional attribute	Description
Priority	Priority that the process runs. Priority values range between -20 (highest) to +19 (lowest). Type and dimension: string-scalar Default: 10
UserName	This attribute is the owner of the process. The process runs with the user ID. Type and dimension: string-scalar Default: root

Resource type definition

```

type Process (
    static keylist SupportedActions = { "program.vfd", getcksum }
    static str ArgList[] = { PathName, Arguments, UserName,
        Priority, PidFile }
    str PathName
    str Arguments
    str UserName = root
    str Priority = 10
    str PidFile
)

```

Sample configurations

Configuration

In this example, the Process agent starts, stops, and monitors sendmail. This process is started with two arguments as determined in the Arguments attribute. The pid stored in the PidFile attribute is used to monitor the sendmail process.

```

Process sendmail (
    PathName = "/usr/sbin/sendmail"
    Arguments = "-bd -q30m"
    PidFile = "/var/run/sendmail.pid"
)

```

ProcessOnOnly agent

Starts, stops, and monitors a process that you specify.

Agent functions

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

State definitions

ONLINE	Indicates that the specified process is running.
FAULTED	Indicates that the process has unexpectedly terminated.
UNKNOWN	Indicates that the agent can not determine the state of the process.

Attributes

Table 5-5 Required attributes

Required attribute	Description
PathName	Defines complete pathname to access an executable program. This path includes the program name. If a process is controlled by a script, the PathName defines the complete path to the shell. The PathName attribute must not exceed 256 characters. Type and dimension: string-scalar

Table 5-6 Optional attributes

Optional attribute	Description
Arguments	<p>Passes arguments to the process. If a process is controlled by a script, the script is passed as an argument. Multiple arguments must be separated by a single space. A string cannot accommodate more than one space between arguments, nor allow for leading or trailing whitespace characters.</p> <p>Type and dimension: string-scalar</p> <p>Example: "-bd -q30m"</p>
IgnoreArgs	<p>A flag that indicates whether monitor ignores the argument list.</p> <ul style="list-style-type: none"> ■ If the value is 0, it checks the process pathname and argument list. ■ If the value is 1, it only checks for the executable pathname and ignores the rest of the argument list. <p>Type and dimension: boolean-scalar</p> <p>Default: 0</p>
PidFile	<p>The file that contains the process ID for the monitoring process. Specify the PidFile attribute for the monitoring process to use the Pid. Otherwise, to complete the monitoring process the agent uses the ps output.</p> <p>Note that when you use scripts, or other indirect mechanisms, to start processes, you must set the PidFile attribute when the ps output is different from the configured values for the PathName or Arguments attributes.</p> <p>Type and dimension: string-scalar</p> <p>Example: "/var/lock/sendmail.pid"</p>
Priority	<p>Priority with which the process will run. Priority values range between -20 (highest) to +19 (lowest).</p> <p>Type and dimension: string-scalar</p> <p>Default: 10</p>

Table 5-6 Optional attributes

Optional attribute	Description
UserName	Owner of the process. The process runs with the user ID. Type and dimension: string-scalar Default: root

Resource type definition

```

type ProcessOnOnly (
    static str ArgList[] = { PathName, Arguments, UserName,
        Priority, PidFile, IgnoreArgs }
    static str Operations = OnOnly
    str PathName
    str Arguments
    str UserName = root
    str Priority = 10
    str PidFile
    boolean IgnoreArgs = 0
)

```

Sample configurations

Configuration 1

```

ProcessOnOnly nfs_daemon(
    PathName = "/usr/lib/nfs/nfsd"
    Arguments = "-a 8"
)

```

Configuration 2

```

include "types.cf"

cluster ProcessCluster (
    .
    .
    .
    group ProcessOnOnlyGroup (
        SystemList = { sysa, sysb }
        AutoStartList = { sysa }
    )

    ProcessOnOnly Process1 (
        PathName = "/usr/local/bin/myprog"

```

```
        Arguments = "arg1 arg2"
    )

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

// resource dependency tree
//
//   group ProcessOnOnlyGroup
//   {
//     ProcessOnOnly Process1
//     ProcessOnOnly Process2
//   }
```

Infrastructure and support agents

This chapter contains:

- [“NotifierMngr agent”](#) on page 62
- [“VRTSWebApp agent”](#) on page 69
- [“Proxy agent”](#) on page 71
- [“Phantom agent”](#) on page 74

About the infrastructure and support agents

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

NotifierMngr agent

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

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

Dependency

The NotifierMngr resource depends on the NIC resource.

Agent functions

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

State definitions

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

Attributes

Table 6-1 Required attributes

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

Table 6-2 Optional attributes

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

Table 6-2 Optional attributes

Optional attribute	Description
MessagesQueue	<p>Size of the VCS engine's message queue. Minimum value is 30.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 30</p>
NotifierListeningPort	<p>Any valid, unused TCP/IP port numbers.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 14144</p>
SmtpFromPath	<p>Set to a valid email address, if you want the notifier to use a custom email address in the FROM: field.</p> <p>Type and dimension: string-scalar</p> <p>Example: "usera@example.com"</p>
SmtpRecipients	<p>Specifies the email address where SMTP sends information and the severity level of the messages. The severity levels of messages are Information, Warning, Error, and SevereError. Specifying a given severity level for messages indicates that all messages of equal or higher severity are received.</p> <p>Note: SmtpRecipients is a required attribute if you specify SmtpServer.</p> <p>Type and dimension: string-association</p> <p>Example:</p> <p>"james@veritas.com" = SevereError, "admin@veritas.com" = Warning</p>

Table 6-2 Optional attributes

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

Table 6-2 Optional attributes

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

Resource type definition

```
type NotifierMngr (  
    static int RestartLimit = 3  
    static str ArgList[] = { EngineListeningPort, MessagesQueue,  
        NotifierListeningPort, SnmpdTrapPort, SnmpCommunity,  
        SnmpConsoles, SntpServer, SntpServerVrfyOff, SntpServerTimeout,  
        SntpReturnPath, SntpFromPath, SntpRecipients }  
    int EngineListeningPort = 14141  
    int MessagesQueue = 30  
    int NotifierListeningPort = 14144  
    int SnmpdTrapPort = 162  
    str SnmpCommunity = public  
    str SnmpConsoles{}  
    str SntpServer  
    boolean SntpServerVrfyOff = 0  
    int SntpServerTimeout = 10  
    str SntpReturnPath  
    str SntpFromPath  
    str SntpRecipients{}  
)
```

Sample configuration

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

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

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

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

The NotifierMngr resource sets up notification for all events to the `SnmpConsole: snmpserv`. In this example, only messages of `SevereError` level are sent to the `SmtpServer (smtp.example.com)`, and the recipient (`vcadmin@example.com`).

Configuration

```
system north

system south

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

Phantom my_phantom (
)

NIC    NicGrp_eth0 (
    Enabled = 1
    Device = eth0
)

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

Proxy nicproxy(
    TargetResName = "NicGrp_eth0"
```

```
)

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

ntfr requires nicproxy

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


Table 6-3 Required attributes

Required attribute	Description
InstallDir	<p>Path to the Web application installation. You must install the Web application as a <code>.war</code> file with the same name as the <code>AppName</code> parameter. Point this attribute to the directory that contains this <code>.war</code> file.</p> <p>Type and dimension: string-scalar</p> <p>Example: If the <code>AppName</code> is <code>cmc</code> and <code>InstallDir</code> is <code>/opt/VRTSweb/VERITAS</code>, the agent constructs the path for the Web application as: <code>/opt/VRTSweb/VERITAS/cmc.war</code></p>
TimeForOnline	<p>The time the Web application takes to start after it is loaded into the Web server. This parameter is returned as the exit value of the online script, which inform VCS of the time it needs to wait before calling monitor on the Web application resource. This attribute value is typically at least five seconds.</p> <p>Type and dimension: integer-scalar</p>

Resource type definition

```
type VRTSWebApp (  
    static int NumThreads = 1  
    static str ArgList[] = { AppName, InstallDir, TimeForOnline }  
    str AppName  
    str InstallDir  
    int TimeForOnline  
)
```

Sample configuration

```
VRTSWebApp VCSweb (  
    AppName = "cmc"  
    InstallDir = "/opt/VRTSweb/VERITAS"  
    TimeForOnline = 5  
)
```

Proxy agent

Mirrors the state of another resource on a local or remote system. Provides a means to specify and modify one resource and have its state reflected by its proxies.

A Proxy resource can only point to None or OnOnly type of resources, and can reside in a failover/parallel group.

Agent functions

Monitor Determines status based on the target resource status.

Attributes

Table 6-4 Required attribute

Required attribute	Description
TargetResName	Name of the target resource that the Proxy resource mirrors. The target resource must be in a different resource group than the Proxy resource. Type and dimension: string-scalar Example: "tmp_VRTSvcs_file1"

Table 6-5 Optional attribute

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

Resource type definition

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

Sample configurations

Configuration 1

The proxy resource mirrors the state of the resource tmp_VRTSvcs_file1 on the local system.

```
Proxy proxy1 (  
    TargetResName = "tmp_VRTSvcs_file1"  
)
```

Configuration 2

The proxy resource mirrors the state of the resource tmp_VRTSvcs_file1 on sysa.

```
Proxy proxy1(  
    TargetResName = "tmp_VRTSvcs_file1"  
    TargetSysName = "sysa"  
)
```

Configuration 3

The proxy resource mirrors the state of the resource mnic on the local system; note that target resource is in grp1, and the proxy is in grp2; a target resource and its proxy cannot be in the same group.

```
group grp1 (  
    SystemList = { sysa, sysb }  
    AutoStartList = { sysa }  
)  
  
MultiNICA mnic (  
    Device @vcslx3 = { eth0 = "192.123.8.42", eth3 =  
        "192.123.8.42" }  
    Device @vcslx4 = { eth0 = "192.123.8.43", eth3 =  
        "192.123.8.43" }  
    NetMask = "255.255.248.0"  
    NetworkHosts = { "192.123.10.129", "192.123.10.130" }  
)
```

```
IPMultiNIC ip1 (  
  Address = "192.123.10.177"  
  MultiNICAResName = mnic  
  NetMask = "255.255.248.0"  
)
```

```
ip1 requires mnic
```

```
group grp2 (  
  SystemList = { sysa, sysb }  
  AutoStartList = { sysa }  
)  
  
IPMultiNIC ip2 (  
  Address = "192.123.10.178"  
  NetMask = "255.255.255.0"  
  MultiNICAResName = mnic  
)  
Proxy proxy (  
  TargetResName = mnic  
)  
ip2 requires proxy
```

Phantom agent

The agent enables VCS to determine the status of parallel service groups that do not include OnOff resources. Do not use the Phantom agent in failover service groups.

Do not attempt manual online or offline operations on the Phantom resource or on the service group containing the Phantom resource. Doing so may result in unpredictable behavior.

Agent functions

Monitor Determines status based on the status of the service group.

Resource type definition

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

Sample configurations

Configuration 1

```
Phantom (  
)
```

Configuration 2

The following example shows a complete main.cf, in which the FileNone resource and the Phantom resource are in the same group.

```
include "types.cf"  
  
cluster PhantomCluster  
  
system sysa  
  
system sysb  
  
group phantomgroup (  
    SystemList = { sysa, sysb }  
    AutoStartList = { sysa }  
    Parallel = 1  
)  
  
FileNone my_file_none (  

```

```
        PathName = "/tmp/file_none"  
    )  
Phantom my_phantom (  
    )  
  
// resource dependency tree  
//  
//     group maingroup  
//     {  
//     Phantom my_Phantom  
//     FileNone my_file_none  
//     }  
// }
```


Testing agents

This chapter contains:

- [“ElifNone agent”](#) on page 78
- [“FileNone agent”](#) on page 79
- [“FileOnOff agent”](#) on page 80
- [“FileOnOnly agent”](#) on page 81

About the program support agents

Use the program support agents to provide high availability for program support resources.

ElifNone agent

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

Agent function

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

Attributes

Table 7-1 Required attribute

Required attribute	Description
PathName	Specifies the complete pathname. Starts with a slash (/) preceding the file name. Type and dimension: string-scalar Example: "/tmp/file01"

Resource type definition

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

Sample configuration

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

FileNone agent

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

Agent functions

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

Attribute

Table 7-2 Required attribute

Required attribute	Description
PathName	Specifies the complete pathname. Starts with a slash (/) preceding the file name. Type and dimension: string-scalar Example: "/tmp/file01"

Resource type definition

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

Sample configuration

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

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	Checks for the specified file. If it exists, the agent reports as ONLINE. If it does not exist, the agent reports as OFFLINE.
Clean	Terminates all ongoing resource actions and takes the resource offline, forcibly when necessary.

Attribute

Table 7-3 Required attribute

Required attribute	Description
PathName	Specifies the complete pathname. Starts with a slash (/) preceding the file name. Type and dimension: string-scalar Example: "/tmp/file01"

Resource type definition

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

Sample configuration

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

FileOnOnly agent

Creates and monitors files.

Agent functions

Online	Creates an empty file with the specified name, unless one already exists.
Monitor	Checks for the specified file. If it exists, the agent reports as ONLINE. If it does not exist, the resource faults.
Clean	Terminates all ongoing resource actions and takes the resource offline, forcibly when necessary.

Attribute

Table 7-4 Required attributes

Required attribute	Description
PathName	Specifies the complete pathname. Starts with a slash (/) preceding the file name. Type and dimension: string-scalar Example: "/tmp/file02"

Resource type definition

```

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

```

Sample configuration

```

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

```


Glossary

administrative IP address

The operating system controls these IP addresses and brings them up even before VCS brings applications online. Use them to access a specific system over the network for doing administrative tasks, for example: examining logs to troubleshoot issues, cleaning up temp files to free space, etc. Typically, you have one administrative IP address per node.

agent function

Agent functions start, stop, fault, forcibly stop, and monitor resources using scripts. Sometimes called an entry point.

base IP address

The first logical IP address, can be used as an administrative IP address.

entry point

See [agent function](#).

floating IP address

See [virtual IP address](#).

logical IP address

Any IP address assigned to a NIC.

NIC bonding

Combining two or more NICs to form a single logical NIC, which creates a fatter pipe.

operation

All agents have scripts that turn the resource on and off. Operations determine the action that the agent passes to the resource. See None operation, OnOff operation, and OnOnly operation.

None operation

For example the NIC resource. Also called persistent resource, this resource is always on. This kind of resource has no online and offline scripts, and only monitors a resource.

OnOff operation

For example the IP and Share agents--in fact most agents are OnOff. This resource has online and offline scripts. Often this type of resource does not appear in the types file because by default when a resource does not have this resource type defined, it is OnOff.

OnOnly operation

For example the NFS, FileOnOnly resources. This kind of resource has an online script, but not an offline one.

plumb

Term for enabling an IP address—used across all platforms in this guide.

test IP address

IP addresses to help determine the state of a link by sending out a ping probe to another NIC (on another system.) Requires a return ping to complete the test. Test IP addresses can be the same as base IP addresses.

virtual IP address

IP addresses that can move from one NIC to another or from one node to another. VCS fails over these IP address with your application. Sometimes called a floating IP address.

Index

A

about

- Network agents 15, 17

agent functions

- Application agent 48
- Disk agent 16
- DNS agent 21
- ElifNone agent 78
- ESXHost agent 44
- ESXVirtualMachine agent 30
- FileNone agent 79
- FileOnOff agent 80
- FileOnOnly agent 81
- GuestOSApp agent 36
- IP agent 18
- NotifierMngr agent 62
- Phantom agent 74
- Process agent 54
- ProcessOnOnly agent 57
- Proxy agent 71
- VMFSVolume agent 38
- VMIP agent 39
- VRTSWebApp agent 69
- VSwitch agent 42

agents

- Application 48
- Disk 16
- DNS 21
- ElifNone 78
- ESXHost 44
- ESXVirtualMachine 30
- FileNone 79
- FileOnOff 80
- FileOnOnly 81
- GuestOSApp 36
- IP 18
- modifying 12
- NotifierMngr 62
- Phantom 74
- Process 54
- ProcessOnOnly 57

- Proxy 71

- typical functions 11

- VMFSVolume 38

- VMIP 39

- VRTSWebApp 69

- VSwitch 42

Application agent

- agent functions 48

- attributes 50

- description 48

- resource type definition 52

- sample configurations 53

- state definitions 49

- association dimension 13

- attribute data types 12

attributes

- Application agent 50

- Disk agent 16

- DNS agent 22

- ElifNone agent 78

- ESXHost agent 44

- ESXVirtualMachine agent 31

- FileNone agent 79

- FileOnOff agent 80

- FileOnOnly agent 81

- GuestOSApp agent 37

- IP agent 19

- modifying 11, 12

- NotifierMngr agent 63

- Process agent 55

- ProcessOnOnly 57

- Proxy agent 71

- VMFSVolume agent 38

- VMIP agent 40

- VRTSWebApp agent 69

- VSwitch agent 42

B

- boolean data types 12

- bundled agents 11

C

- Cluster Manager (Java Console), modifying
 - attributes 12
- Cluster Manager (Web Console)
 - modifying attributes 12
- CNAME record 23
- configuration files
 - main.cf 74
 - modifying 12
 - types.cf 11

D

- data types
 - boolean 12
 - integer 12
 - string 12
- dependencies
 - ESXVirtualMachine agent 30
 - GuestOSApp agent 36
- descriptions
 - resources 11
- dimensions
 - keylist 13
 - scalar 13
 - vector 13
- Disk agent
 - agent functions 16
 - attributes 16
 - description 16
 - resource type definition 16
 - state definitions 16
- DNS agent 21
 - agent functions 21
 - attributes 22
 - description 21
 - Linux attributes 22
 - prerequisites 26
 - resource type definition 23
 - sample web server configuration 25

E

- ElifNone agent
 - agent functions 78
 - attributes 78
 - description 78
 - resource type definition 78
 - sample configuration 78

- ESXHost agent
 - agent functions 44
 - attribute 44
 - description 44
 - resource type definition 45
- ESXVirtualMachine agent
 - agent functions 30
 - attributes 31
 - dependencies 30
 - description 30
 - migration verification 34
 - requirements 30
 - resource type definition 34
 - sample configurations 35
 - state definitions 31

F

- FileNone agent
 - agent functions 79
 - attribute 79
 - description 79
 - resource type definition 79
 - sample configurations 79
- FileOnOff agent
 - agent functions 80
 - attribute 80
 - description 80
- FileOnOnly agent
 - agent functions 81
 - attribute 81
 - description 81
 - resource type definition 81
 - sample configuration 81

G

- GuestOSApp agent
 - agent functions 36
 - attributes 37
 - dependencies 36
 - description 36
 - requirements 36
 - resource type definition 37
 - state definitions 36

I

- integer data types 12

IP agent

- agent functions 18
- attributes 19
- description 18
- resource type definitions 20
- sample configurations 20
- state definitions 18

K

keylist dimension 13

M

- main.cf 11, 74
- migration verification
 - ESXVirtualMachine agent 34
- modifying
 - agents 12
 - attributes 11, 12
 - Cluster Manager (Web Console) 12
 - configuration files 12
- monitor scenarios, DNS agent 25

N

- NotifierMngr agent
 - agent functions 62
 - attributes 63
 - description 62
 - resource type definition 66
 - sample configurations 67
 - state definitions 62

O

online query 23

P

- Phantom agent
 - agent functions 74
 - description 74
 - resource type definition 74
 - sample configurations 74

Process agent

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

ProcessOnOnly agent

- agent functions 57
- attributes 57
- description 57
- resource type definition 59
- sample configurations 59
- state definitions 57

Proxy agent

- agent functions 71
- attributes 71
- description 71
- resource type definition 72
- sample configurations 72

R

- requirements
 - ESXVirtualMachine agent 30
 - GuestOSApp agent 36
 - VMIP agent 39
- resource type definitions
 - Application agent 52
 - Disk agent 16
 - DNS agent 23
 - ElifNone agent 78
 - ESXHost agent 45
 - ESXVirtualMachine agent 34
 - FileNone agent 79
 - FileOnOnly agent 81
 - GuestOSApp agent 37
 - IP agent 20
 - NotifierMngr agent 66
 - Phantom agent 74
 - Process agent 56
 - ProcessOnOnly agent 59
 - Proxy agent 72
 - VMFSVolume agent 38
 - VMIP agent 41
 - VRTSWebApp agent 70
 - VSwitch agent 43
- resource types 11
- resources, description of 11

S

sample configurations

- Application agent 53
- ElifNone agent 78
- ESXVirtualMachine agent 35
- FileNone agent 79
- FileOnOff agent 80
- FileOnOnly agent 81
- IP agent 20
- NotifierMngr agent 67
- Phantom agent 74
- Process agent 56
- ProcessOnOnly agent 59
- Proxy agent 72
- VMFSVolume agent 38
- VMIP agent 41
- VRTSWebApp agent 70
- VSwitch agent 43

sample DNS configuration 26

scalar dimension 13

state definitions 21

- Application agent 49
- Disk agent 16
- DNS agent 21
- ESXVirtualMachine agent 31
- GuestOSApp agent 36
- IP agent 18
- NotifierMngr agent 62
- Process agent 54
- ProcessOnOnly agent 57
- VMFSVolume agent 38
- VMIP agent 39
- VRTSWebApp agent 69
- VSwitch agent 42

string data type 12

T

types.cf 11

V

VCS, resource types 11

vector dimension 13

VMFSVolume agent

- agent functions 38
- attribute 38
- description 38
- resource type definition 38
- sample configurations 38

state definition 38

VMIP agent

- agent functions 39
- attributes 40
- description 39
- requirements 39
- resource type definition 41
- sample configurations 41
- state definitions 39

VRTSWebApp agent

- agent functions 69
- attributes 69
- description 69
- resource type definition 70
- sample configuration 70
- state definitions 69

VSwitch agent

- agent functions 42
- attributes 42
- description 42
- resource type definitions 43
- sample configurations 43
- state definitions 42