

Symantec ApplicationHA Bundled Agents Guide

Windows on Hyper-V

6.1

Symantec™ ApplicationHA Bundled Agents Guide

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

Product_version: 6.1

Document_version: 6.1 Rev 0

Legal Notice

Copyright © 2014 Symantec Corporation. All rights reserved.

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

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

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

The Licensed Software and Documentation are deemed to be commercial computer software as defined in FAR 12.212 and subject to restricted rights as defined in FAR Section 52.227-19 "Commercial Computer Software - Restricted Rights" and DFARS 227.7202, "Rights in Commercial Computer Software or Commercial Computer Software Documentation", as applicable, and any successor regulations, whether delivered by Symantec 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.

Symantec Corporation
350 Ellis Street
Mountain View, CA 94043

<http://www.symantec.com>

Technical Support

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

Symantec's support offerings include the following:

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

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

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

All support services will be delivered in accordance with your support agreement and the then-current enterprise technical support policy.

Contacting Technical Support

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

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

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

When you contact Technical Support, please have the following information available:

- Product release level
- Hardware information

- Available memory, disk space, and NIC information
- Operating system
- Version and patch level
- Network topology
- Router, gateway, and IP address information
- Problem description:
 - Error messages and log files
 - Troubleshooting that was performed before contacting Symantec
 - Recent software configuration changes and network changes

Licensing and registration

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

www.symantec.com/business/support/

Customer service

Customer service information is available at the following URL:

www.symantec.com/business/support/

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

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

Support agreement resources

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

Asia-Pacific and Japan customercare_apac@symantec.com

Europe, Middle-East, and Africa semea@symantec.com

North America and Latin America supportsolutions@symantec.com

About Symantec Connect

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

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

Documentation

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

doc_feedback@symantec.com

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

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

Introducing Symantec ApplicationHA bundled agents

This document includes the following topics:

- [About ApplicationHA agents](#)
- [About the Heartbeat agent](#)
- [About the MountMonitor agent](#)
- [About the ServiceMonitor agent](#)
- [About the GenericService agent](#)
- [About the Process agent](#)
- [About the FileNone agent](#)
- [About the ElifNone agent](#)
- [About the FileOnOff agent](#)
- [About the FileOnOnly agent](#)

About ApplicationHA agents

Agents are application-specific modules that plug into the ApplicationHA framework that manages the components of the configured applications.

The agents are installed when you install ApplicationHA. These agents start, stop, and monitor the components of the configured applications and report its state

changes. If an application or its components fail, these agents restart the applications and its components on the virtual machine.

A virtual machine has one agent per component that monitors all the components of that type. For example, a single GenericService agent manages all services that are configured using the GenericService components. When the agent starts, it obtains the necessary configuration information from these components and then monitors the configured applications. The agents then periodically updates ApplicationHA with the component and application status.

Agents perform the following operations:

- Brings the components online
- Takes the components offline
- Monitors the components and reports the state changes

ApplicationHA agents are classified in the following categories:

- Infrastructure agents (bundled agents)
Infrastructure agents are packaged (bundled) with the base software and include agents for mount points, generic services and processes. These agents are immediately available for use after you install ApplicationHA.
- Application agents
Application agents are used to monitor third party applications such as Microsoft SQL Server, Microsoft Exchange and so on. These agents are packaged separately and are available in the form of an agent pack that gets installed when you install ApplicationHA.
The agent pack is released on a quarterly basis. The agent pack includes support for new applications as well as fixes and enhancements to existing agents. You can install the agent pack on an existing ApplicationHA installation.
Refer to the Symantec Operations Readiness Tools (SORT) Website for information on the latest agent pack availability:

<https://sort.symantec.com>

This document describes the ApplicationHA bundled agents along with their resource type definitions, attribute definitions, and sample configurations.

About intelligent monitoring framework

ApplicationHA provides Intelligent Monitoring Framework (IMF) to determine the status of the configured application and its components. IMF employs an event-based monitoring framework that is implemented using custom as well as native operating system-based notification mechanisms.

IMF provides instantaneous state change notifications. ApplicationHA agents detect this state change and then trigger the necessary actions.

IMF provides the following key benefits:

- Instantaneous notification
Faster fault detection resulting in faster fail over and thus less application down time.
- Ability to monitor large number of components
With reduced CPU consumption, IMF effectively monitors a large number of components.
- Reduction in system resource utilization
Reduced CPU utilization by ApplicationHA agent processes when number of components being monitored is high. This provides significant performance benefits in terms of system resource utilization.

How IMF works

The following steps outline how IMF-based monitoring works:

1. When IMF is enabled, the ApplicationHA agent waits for the components to report the same steady state (whether online or offline) for two consecutive monitor cycles and then registers the components for IMF-based monitoring.
2. The agent then registers itself for receiving specific custom or operating system specific event notifications.
3. In case of an application failure, the agent determines the affected component and then executes a monitor cycle for that component. The monitor cycle determines the component status. If the component state is offline, then ApplicationHA takes the necessary corrective action, depending on the configuration.
4. If the component state remains the same, then the agent moves to a wait state and then waits for the next event to occur.

About the agent functions and attributes

Every agent has a collection of attributes and performs a definite set of functions.

Attributes are the set of variables whose values configures the corresponding application component to function in a specific way. By modifying attribute values you can change the way in which ApplicationHA agent manages the component.

For example, the IP agent monitors an IP address. The specific address to be monitored is identified by the attribute "Address" whose value is the specific IP address.

Depending on the category to which an agent belongs, an agent performs either or all of the following functions:

Online	Brings the configured component online
Offline	Takes the configured component offline
Monitor	Verifies if the configured component is online

As part of the Monitor function, an agent reports the following states:

ONLINE	Indicates that the configured component is online
OFFLINE	Indicates that the configured component/application has faulted
UNKNOWN	Indicates that the agent encountered errors while monitoring the configured component

About the Heartbeat agent

The Application Monitoring Heartbeat Agent for Microsoft Hyper-V (HyperVAppMonHB) monitors the configured application and provides the application status to the Hyper-V host in form of an heartbeat. If the application runs without a fault, the Heartbeat agent indicates the state as “Applications Healthy”. If any one or all the components of the application faults, then the agent indicates the application state as “Applications Critical”. This marks an application failure and the Hyper-V host takes the corrective action as per the configured policies.

While configuring application monitoring, the application configuration wizard creates a resource for the Heartbeat agent. This resource verifies the status of the configured application. A single resource is used to monitor all components of the configured application.

Note: No separate resources are created, if you configure additional applications, using the command line. The single resource that is created monitors all the components, even if multiple applications are configured.

While configuring multiple applications, you must include the names of all the component dependency groups created, in the ServiceGroupName attribute.

The heartbeat agent is represented by the HyperVAppMonHB resource type.

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.

Agent functions

Monitor Verifies that the configured application is online.

Agent state definitions

ONLINE Indicates that the configured application is online.

OFFLINE Indicates that the components of the configured application have faulted.

UNKNOWN Indicates that the agent encountered errors while monitoring the application or communicating with the Hyper-V host.

Resource type definition

```
type HyperVAppMonHB (
  static int IMF{} = { Mode=2, MonitorFreq=30, RegisterRetryLimit=3 }
  static i18nstr IMFRegList[] = { ServiceGroupName }
  static int MonitorInterval = 10
  static int OfflineMonitorInterval = 10
  static int NumThreads = 1
  static i18nstr ArgList[] = { DelayBeforeAppFault, ServiceGroupName,
  FaultPolicy, VMGracefulRebootAttempts, VMGracefulRebootTimeSpan,
  VMGracefulRebootPolicy }
  static str Operations = None
  str ServiceGroupName[]
  int DelayBeforeAppFault = 300
  str FaultPolicy = ONE
  int VMGracefulRebootAttempts = 1
  int VMGracefulRebootTimeSpan = 1
  int VMGracefulRebootPolicy = 0
)
```

Agent attributes

[Table 1-1](#) describes the Heartbeat agent attributes.

Table 1-1 Heartbeat agent attributes

Attribute	Description
ServiceGroupName	<p>Defines the name of the component dependency group that is created for the application being monitored. The component dependency group name is created when you run the application monitoring configuration wizard.</p> <p>If you configure additional applications, using the command line, you must include the names of all those component dependency groups in this attribute.</p>
DelayBeforeAppFault	<p>Defines the time (in seconds) for which the agent must wait for all the application components to go offline. After all the components are offline, the agent indicates the application state “Applications Critical” to the Hyper-V host.</p> <p>This attribute can take a value between 0 to 600 seconds. Default is 300 seconds.</p>
FaultPolicy	<p>Defines the state of the application based on the number of components faulted. It contains the following values:</p> <ul style="list-style-type: none"> ■ ONE: If the value is set to ONE, the agent indicates the application state “Applications Critical” even if one of the component faults. ■ ALL: If the value is set to ALL, the agent indicates the application state “Applications Critical” only if all the components fault. <p>The default value is ONE.</p>

Table 1-1 Heartbeat agent attributes (*continued*)

Attribute	Description
VM.GracefulRebootPolicy	<p>Use this option to enable or disable ApplicationHA-initiated virtual machine restart policy. This option defines whether or not ApplicationHA restarts the virtual machine in response to application and component failures.</p> <p>When a configured application or component fails, ApplicationHA attempts to restart the failed components. If the component fails to start, ApplicationHA then does not take any action.</p> <p>If this policy is disabled, when a configured application or component fails, ApplicationHA attempts to restart the failed components. If the component fails to start, ApplicationHA sends a "Applications Critical" status to the Hyper-V host. The Hyper-V host then takes the necessary action as per the configuration settings.</p> <p>VM.GracefulRebootPolicy value can be Enabled (1) or Disabled (0). The default value is Disabled.</p>
VM.GracefulRebootAttempts	<p>This option defines the number of times ApplicationHA attempts to restart the virtual machine gracefully if the configured application or component becomes unresponsive. The number of restart attempts is time bound and is defined by the option VM.GracefulRebootTimeSpan. The restart attempts count is reset after the reboot time span elapses.</p> <p>For example, if the reboot attempts value is 4, the time span value is 60 minutes, and ApplicationHA has restarted the virtual machine once, then the restart attempt count is 3 (initial set value of 4 minus one reboot) for the remaining period of the 60-minute interval. The restart attempts count is reset to 4 at the beginning of the next 60-minute span.</p> <p>If the restart attempts are exhausted ApplicationHA does not take any action.</p> <p>VM.GracefulRebootAttempts value can be between 1 and 10. The default value is 1.</p>

Table 1-1 Heartbeat agent attributes (*continued*)

Attribute	Description
VM.GracefulRebootTimeSpan	<p>This option defines the time interval, in hours, during which ApplicationHA can gracefully restart the virtual machine for the number of times defined by the option VM.GracefulRebootAttempts.</p> <p>This attribute is applicable only if the attribute VM.GracefulRebootPolicy is set to 1.</p> <p>VM.GracefulRebootTimeSpan value can be between 1 and 24. The default value is 1 hour.</p>

About the MountMonitor agent

The MountMonitor agent monitors the mount path of the configured storage. It is independent of how the underlying storage is managed. The mount path can be a drive letter or a folder mount.

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

This agent is represented by the MountMonitor resource type.

Agent functions

Online	Mounts the configured mount path (drive letter or folder) on the virtual machine.
Monitor	Verifies that the specified mouth path (drive letter or folder) is mounted.

Agent state definitions

ONLINE	Indicates that the virtual machine can access the configured mount path.
OFFLINE	Indicates that the virtual machine cannot access the configured mount path.
UNKNOWN	Indicates that the agent cannot determine the status of the component (resource).

Agent resource type definition

```

type MountMonitor (
  static int IMF{} = { Mode=3, MonitorFreq=5, RegisterRetryLimit=3 }
  static il8nstr IMFRegList[] = { MountPath, VolumeName }
  static il8nstr ArgList[] = { MountPath, VolumeName, MountDependsOn }
  static str Operations = OnOnly
  str MountPath
  str VolumeName
  str MountDependsOn{}
)

```

Agent attributes

[Table 1-2](#) describes the MountMonitor agent attributes.

Table 1-2 MountMonitor agent required attributes

Required attributes	Description
MountPath	<p>The drive letter or path to an empty NTFS folder where a partition is mounted. The attribute can be specified as X, X:, X:\, X:\Directory, X:\Directory\.</p> <p>When configuring a directory to host the mount, verify the following conditions:</p> <ul style="list-style-type: none"> ■ The configured path exists ■ The directory is empty ■ The volume on which the directory resides is NTFS-formatted <p>Type and dimension: string-scalar</p>
VolumeName	The GUID of the volume to be mounted.

Table 1-2 MountMonitor agent required attributes (*continued*)

Required attributes	Description
MountDependsOn	<p>Defines the dependency between the nested mount points.</p> <p>If the application data is stored on nested mount points, then it is required to set the dependency between these mount points. This dependency enables ApplicationHA to monitor all the nested mount points.</p> <p>If this attribute is not configured, then ApplicationHA monitors only the last mount point.</p> <p>The value of this attribute is a key-value pair.</p> <p>Where,</p> <p>Key= mount path</p> <p>Value= volume name</p>

About the 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 Service Control Manager (SCM). The agent can be configured to monitor a service using the 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 virtual machine, once every monitor cycle. The agent returns ONLINE, OFFLINE, or UNKNOWN depending on the exit code of the script. You can monitor a service or run a script in a user-context by specifying the user name, password, and domain.

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

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.

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.

Agent state definitions

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

Agent resource type definition

The Service Monitor agent is represented by the ServiceMonitor resource type.

```
type ServiceMonitor (
  static int IMF{} = { Mode=3, MonitorFreq=5, RegisterRetryLimit=3 }
  static i18nstr IMFRegList[] = { ServiceOrScriptName, MonitorService }
  static int OfflineMonitorInterval = 60
  static i18nstr ArgList[] = { MonitorService, ServiceOrScriptName,
  MonitorProgTimeout, UserAccount, Password, Domain }
  static str Operations = None
  boolean MonitorService = 1
  i18nstr ServiceOrScriptName
  int MonitorProgTimeout = 30
  i18nstr UserAccount
  str Password
  i18nstr Domain
)
```

Agent attributes

[Table 1-3](#) describes the ServiceMonitor agent required attributes.

Table 1-3 ServiceMonitor agent required attributes

Required Attribute	Description
ServiceOrScriptName	<p>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.</p> <p>This attribute can take localized values.</p> <p>Note: The exit code of the script must conform to ApplicationHA conventions: 110 for ONLINE and 100 for OFFLINE. For exit values outside the range 100-110, the status is considered UNKNOWN.</p> <p>Type and dimension: string-scalar</p>

Table 1-4 describes the ServiceMonitor agent optional attributes.

Table 1-4 ServiceMonitor agent optional attributes

Optional Attributes	Description
Domain	<p>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 computer.</p> <p>Note: This attribute can take localized values.</p>
MonitorProgTimeout	<p>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.</p> <p>Default is 30 seconds.</p>
MonitorService	<p>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.</p> <p>Default is 1.</p>
Password	<p>The password for the user account, encrypted using the ApplicationHA Encrypt utility. This attribute is valid only if the MonitorService is set to 0 and UserAccount is not empty.</p>

Table 1-4 ServiceMonitor agent optional attributes (*continued*)

Optional Attributes	Description
UserAccount	<p>A valid user account in whose context the service will be monitored. Username can be of the form <i>username@domain.com</i> or <i>domain.com\username</i></p> <p>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.</p> <p>For example: On Windows 2003: User Account="LocalSystem", "Local Service", or "Network Service". Domain="NT Authority".</p> <p>The 'NT Authority' domain is not applicable for the 'LocalSystem' account.</p> <p>Note: This attribute can take localized values.</p>

About the GenericService agent

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

Services are configured 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.

Note: The service to be configured using the GenericService agent must not be in a disabled state.

This agent is represented by the GenericService resource type.

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.

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.

Agent state definitions

ONLINE Indicates that the service being monitored is running.

OFFLINE Indicates that the service being monitored is stopped.

UNKNOWN Indicates the service operation is in a pending state, or that the agent cannot determine the state of the resource.

Agent resource type definition

```
type GenericService (
  static int IMF() = { Mode=3, MonitorFreq=5, RegisterRetryLimit=3 }
  static i18nstr IMFRegList[] = { ServiceName }
  static i18nstr ArgList[] = { ServiceName, DelayAfterOnline,
  DelayAfterOffline, UserAccount, Password, Domain, service_arg,
  UseVirtualName, "LanmanResName:VirtualName" }
  i18nstr ServiceName
  int DelayAfterOnline = 10
  int DelayAfterOffline = 10
  i18nstr UserAccount
  str Password
  i18nstr Domain
  str service_arg[]
  boolean UseVirtualName = 0
)
```

Agent attributes

[Table 1-5](#) describes the GenericService agent required attributes.

Table 1-5 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. Type and dimension: string-scalar

[Table 1-6](#) describes the GenericService agent required attributes.

Table 1-6 GenericService agent required attributes

Optional attributes	Description
DelayAfterOffline	<p>Number of seconds the offline routine waits for the service to go offline.</p> <p>Default is 10 seconds.</p> <p>Type and dimension: integer-scalar</p>
DelayAfterOnline	<p>Number of seconds the online routine waits for the service to go online.</p> <p>Default is 10 seconds.</p> <p>Type and dimension: integer-scalar</p>
Domain	<p>The domain name to which the user specified in the UserAccount attribute belongs.</p> <p>If the UserAccount attribute is empty or contains a built-in service account, this attribute is ignored.</p> <p>Type and dimension: string-scalar</p>
Password	<p>The password of the user, in whose context, the service would be started.</p> <p>Encrypt the password using the VCSEncrypt utility.</p> <p>If the UserAccount attribute is empty or contains a built-in service account, this attribute is ignored.</p> <p>Type and dimension: string-scalar</p>
service_arg	<p>An array of arguments passed to the service.</p> <p>Type and dimension: string-vector</p>

Table 1-6 GenericService agent required attributes (*continued*)

Optional attributes	Description
UserAccount	<p>A valid user account in whose context the service will be monitored. The user name can be of the form username@domain.com or domain.com\username.</p> <p>If the startup type of the configured service is set to Automatic, then the user account you specify here must be the same as that specified in the Windows Service Control Manager (SCM).</p> <p>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.</p> <p>For example:</p> <p>On Windows 2003: UserAccount='LocalSystem', 'Local Service', or 'Network Service'. Domain='NT Authority'.</p> <p>The 'NT Authority' domain is not applicable for the 'LocalSystem' account.</p> <p>Type and dimension: string-scalar</p>

About the Process agent

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

This agent is represented by the Process resource type.

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.

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 cannot determine the status of the resource.

Resource type definition

```

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

```

Agent attributes

[Table 1-7](#) describes the Process agent required attributes.

Table 1-7 Process agent required attributes

Required attribute	Description
StartProgram	<p>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.</p> <p>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.</p> <p>If you define the start program as a script (a batch file, a Perl script, or a vbs script), the start program should be the program that interprets the script (cmd.exe, or perl.exe, or cscript.exe) and the script itself should be passed as an argument.</p> <p>Type and dimension: string-scalar</p>

[Table 1-8](#) describes the Process agent optional attributes.

Table 1-8 Process agent optional attributes

Optional attributes	Description
CleanProgram	<p>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.</p> <p>Type and dimension: string-scalar</p>
CleanProgramTimeout	<p>The maximum time, in seconds, that the agent must wait before killing the process specified in the CleanProgram attribute.</p> <p>This attribute is ignored if the clean program is not specified.</p> <p>The default value is 30 seconds.</p> <p>Type and dimension: integer-scalar</p>
Domain	<p>The domain in which the user specified by the attribute UserName exists. This attribute is ignored if the user name is not specified.</p> <p>Type and dimension: string-scalar</p>

Table 1-8 Process agent optional attributes (*continued*)

Optional attributes	Description
InteractWithDesktop	<p>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.</p> <p>Default is 0.</p> <p>Type and dimension: boolean-scalar</p>
MonitorProgram	<p>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.</p> <p>If you do not specify a value for this attribute, Symantec ApplicationHA 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.</p> <p>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.</p> <p>Type and dimension: string-scalar</p> <p>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 Symantec ApplicationHA conventions: 110 for ONLINE and 100 for OFFLINE. For exit values outside the range 100-110, the status is considered UNKNOWN.</p>
MonitorProgramTimeout	<p>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.</p> <p>Default is 30 seconds.</p> <p>Type and dimension: integer-scalar</p>

Table 1-8 Process agent optional attributes (*continued*)

Optional attributes	Description
Password	<p>The encrypted password of the user specified by the UserName.</p> <p>Note that the password must be encrypted using the VCSEncrypt utility.</p> <p>This attribute is ignored if the user name is not specified.</p> <p>Type and dimension: string-scalar</p>
StartupDirectory	<p>The startup directory for the process indicated by the StartProgram attribute.</p> <p>Type and dimension: string-scalar</p>
StopProgram	<p>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.</p> <p>If you do not specify a value for this attribute, Symantec ApplicationHA stops the start program.</p> <p>Type and dimension: string-scalar</p> <p>Note: If successful, the StopProgram returns a positive value. The Monitor routine is called 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.</p> <p>Type and Dimension: string-scalar</p>
StopProgramTimeout	<p>The maximum time, in seconds, that the agent must wait before killing the process specified in the StopProgram attribute.</p> <p>The default value is 30 seconds.</p> <p>Type and dimension: integer-scalar</p>
UserName	<p>The user name with whose privileges the configured process executes. User name can be of the form <i>username@domain.com</i> or <i>domain.com\username</i>.</p> <p>If a user name is not specified, the configured process runs in the context of the local system account.</p> <p>Type and dimension: string-scalar</p>

Note: When defining the StartProgram, StopProgram, or MonitorProgram attributes, enclose the path of the executable file and its arguments in double quotes.

About the FileNone agent

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

This agent is represented by the FileNone resource type.

Agent function

Monitor Verifies that the specified file exists

Agent 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 valid file name

Resource type definition

```
type FileNone (
  static i18nstr ArgList[] = { PathName }
  static int OfflineMonitorInterval = 60
  static str Operations = None
  i18nstr PathName )
```

Agent attribute

[Table 1-9](#) describes the FileNone agent required attributes.

Table 1-9 FileNone agent required attribute

Required Attributes	Description
PathName	The complete path of the file to be monitored. Type and dimension: string-scalar

About the ElifNone agent

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

This agent is represented by the ElifNone resource type.

Agent function

Monitor Verifies that the specified file exists

State definitions

ONLINE Indicates the specified file does not exist

OFFLINE Indicates the specified file exists

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

Resource type definition

```
type ElifNone (
static i18nstr ArgList[] = { PathName }
static int OfflineMonitorInterval = 60
static str Operations = None
i18nstr PathName
)
```

Agent attribute

[Table 1-10](#) describes the ElifNone agent required attributes.

Table 1-10 ElifNone agent required attribute

Required attributes	Description
PathName	The complete path of the file to be monitored. Type and dimension: string-scalar

About the FileOnOff agent

The FileOnOff agent creates, removes, and monitors files.

This agent is represented by the FileOnOff resource type.

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 valid file name.

Resource type definition

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

Agent attribute

[Table 1-11](#) describes the FileOnOff agent required attribute.

Table 1-11 FileOnOff agent required attribute

Required Attributes	Description
PathName	The complete path of the file to be monitored. Type and dimension: string-scalar

About the FileOnOnly agent

The FileOnOnly agent creates and monitors a file.

This agent is represented by the FileOnOnly resource type.

Agent functions

Online	Creates 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 valid file name

Resource type definition

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

Agent attribute

[Table 1-12](#) describes the FileOnOnly agent required attribute.

Table 1-12 FileOnOnly agent required attribute

Required attributes	Description
PathName	The complete path of the file to be monitored. Type and dimension: string-scalar

Index

A

- about
 - agent functions and attributes 9
 - ApplicationHA agents 7
 - ApplicationHA agents; IMF 8
 - ElifNone agent 28
 - FileNone agent 27
 - FileOnOff agent 28
 - FileOnOnly agent 29
 - GenericService agent 19
 - Heartbeat agent 10
 - intelligent monitoring framework 8
 - MountMonitor agent 14
 - Process agent 22
 - ServiceMonitor agent 16
- agent attributes
 - GenericService 20
 - Heartbeat agent 11
 - MountMonitor 15
 - Process agent 23
 - ServiceMonitor 17
- agent function
 - MountMonitor 14
- agent functions
 - GenericService 19
 - Heartbeat agent 11
 - Process agent 22
 - ServiceMonitor 16
- agent resource type definition
 - MountMonitor 15
- agent state definition
 - MountMonitor 14

E

- ElifNone
 - agent attributes 28
 - agent function 28
 - resource type definition 28
 - state definitions 28

F

- FileNone
 - agent attribute 27
 - agent function 27
 - resource type definition 27
 - state definitions 27
- FileOnOff
 - agent attribute 29
 - agent functions 28
 - resource type definition 29
 - state definitions 29
- FileOnOnly
 - agent attribute 30
 - agent functions 29
 - resource type definition 30
 - state definitions 29

G

- GenericService
 - agent attributes 20
 - agent functions 19
 - resource type definition 20
 - state definitions 20

H

- Heartbeat agent
 - agent attributes 11
 - agent functions 11
 - resource type definition 11
 - state definitions 11

M

- MountMonitor
 - agent function 14
 - agent required attributes 15
 - agent resource type definition 15
 - state definition 14

P

- Process agent
 - agent attributes 23
 - agent functions 22
 - resource type definition 23
 - state definitions 23

R

- resource type definition
 - GenericService 20
 - Heartbeat agent 11
 - Process agent 23
 - ServiceMonitor 17

S

- ServiceMonitor
 - agent attributes 17
 - agent functions 16
 - resource type definition 17
 - state definitions 17
- state definitions
 - GenericService 20
 - Heartbeat agent 11
 - Process agent 23
 - ServiceMonitor 17