Symantec™ ApplicationHA 6.2 Agent for Apache HTTP Server Configuration Guide - Linux on KVM

November 2014

Symantec

Symantec™ ApplicationHA Agent for Apache HTTP Server Configuration Guide

Agent version: 6.2

Document version: 6.2 Rev 1

Legal Notice

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_apj@symantec.com
Europe, Middle-East, and Africa	semea@symantec.com
North America and Latin America	supportsolutions@symantec.com

Documentation

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

https://sort.symantec.com/documents

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

doc_feedback@symantec.com

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

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

About Symantec Connect

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

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

Contents

Chapter 1

Introducing the Symantec ApplicationHA Agent for Apache HTTP Server

This chapter includes the following topics:

About the Symantec agent for Apache HTTP Server

The Symantec ApplicationHA agents monitor specific components within an enterprise application. They determine the status of the application instances and start or stop them according to external events.

The Symantec ApplicationHA agent for Apache HTTP Server provides high availability for Apache HTTP Server instances.

IMF awareness

The Apache HTTP Server agent is Intelligent Monitoring Framework (IMF)-aware and uses asynchronous monitoring framework (AMF) kernel driver for IMF notification. If you use the Symantec ApplicationHA Configuration Wizard to configure the agent, the wizard by default enables IMF.

For more information on IMF and intelligent resource monitoring, refer to the *Symantec Cluster Server Administrator's Guide*.

For more information about IMF-related Apache HTTP Server agent functions: See "Apache HTTP Server agent functions " on page 12.

Using Apache HTTP Server agent with IMF

The Apache agent supports Intelligent Monitoring Framework only during the process online operation (that is, in the agent's PRON mode).

The agent registers the following two processes for monitoring Apache HTTP server instances with IMF:

- Process with the parent PID init.
- Child process with the maximum elapsed time.

About installing and removing the ApplicationHA agent for Apache HTTP Server

When you install or uninstall Symantec ApplicationHA, the ApplicationHA agent for Apache HTTP Server is automatically installed or removed. For more information, see the *Symantec ApplicationHA Installation and Upgrade Guide*.

When you run the installer or uninstall program that accompanies the quarterly agent pack release of high availability agents from Symantec, the latest version of the ApplicationHA agent for Apache HTTP Server is automatically installed or removed. For more information, see the *Symantec ApplicationHA Agent Pack Installation Guide*.

Supported software

The Symantec ApplicationHA agent for Apache HTTP Server supports the following software versions:

- Symantec ApplicationHA agent for Apache HTTP Server can be installed and run inside virtual machines that have Symantec ApplicationHA 6.2 installed.
- The following versions of the Veritas Operations Manager components are supported:
  - Veritas Operations Manager Management Server 6.0 or later
  - Veritas Operations Manager managed host for Linux: 6.0 or later

Supported application versions

Table 1-1 lists the Apache HTTP Server versions that Symantec ApplicationHA 6.2 currently supports on virtual machine.

Table 1-1 Supported application versions

Application	Version
Apache HTTP Server	■ 1.3, 2.0, and 2.2. Also supports the IBM HTTP Server 7.x.

Supported virtualization environments

Symantec ApplicationHA can be installed and run inside virtual machines in a KVM virtualization environment, running Red Hat Enterprise Linux (RHEL) 6, Update 3 and 4 in the physical host.

Supported operating systems on virtual machines

Table 1-2 shows the supported operating systems for Symantec ApplicationHA 6.2.

Table 1-2 Supported guest operating systems

Operating systems	Levels	Kernel version
Red Hat Enterprise Linux 6	Updates 3, 4, and 5	2.6.32-279.el6 2.6.32-358.el6 2.6.32-431
Red Hat Enterprise 7	-	3.10.0-123

Note: Only 64-bit operating systems are supported.

If your system is running a lower level of Red Hat Enterprise Linux, than indicated in Table 1-2, you must upgrade it before attempting to install Symantec ApplicationHA. Consult the Red Hat documentation for more information on upgrading or reinstalling your operating system.

Symantec supports only Red Hat distributed kernel binaries.

Symantec products operate on subsequent kernel and patch releases provided the operating systems maintain kernel ABI (application binary interface) compatibility.

Apache HTTP Server agent functions

The agent consists of resource type declarations and agent executables. The agent executables are organized into online, offline, monitor, and clean functions.

Online

When you click **Start Application**, ApplicationHA performs the following Online tasks :

- Starts the Apache HTTP Server by executing the Apache HTTP Server executable provided when configuring the application, with the appropriate arguments.
- If the EnvFile attribute is configured, the file is sourced before the agent executes the Apache HTTP Server executable.

Offline

When you click **Stop Application**, ApplicationHA performs the following Offline tasks :

- Stops the Apache HTTP Server by executing the Apache HTTP Server executable provided when configuring the application, with the appropriate arguments.
- For Apache v1.3, sends a TERM signal to the HTTP Server parent process.
- If the EnvFile attribute is configured, the file is sourced before the agent executes the Apache HTTP Server executable.

Monitor

This function monitors the state of the Apache HTTP Server instances running in a virtual machine, by performing the following tasks:

- Conducts a first level check, to ensure that all the processes of an Apache HTTP Server instance is running.
- The processes of an Apache HTTP Server instance is identified by applying the pattern matching on command lines of processes running in the virtual machine.
- Depending upon the value of the MonitorProgram attribute, the monitor function can perform an optional check on the Apache HTTP Server instance by using the `ab` utility (Apache benchmarking utility).

Note: To configure second level monitoring, use CLI.

ApplicationHA wizards configure Apache HTTP Server agent for basic or first level monitoring. To enable detailed or second level monitoring, use CLI/Veritas Operation Manager (VOM).

For more information on VCS commands, refer to Symantec Cluster Server documentation.

Also, for more information on detailed monitoring, See "Setting up detail monitoring for a Apache HTTP Server instance" on page 31.

Clean

The clean function performs the following tasks:

- Removes the Apache HTTP Server system resources that may remain after a server fault or after an unsuccessful attempt to start or stop the application. These resources include the parent Apache HTTP Server processes and its child processes.

imf_init

This function performs the following task:

- Initializes the agent to interface with the AMF kernel driver. This function runs when the agent starts.

imf_getnotification

This function performs the following task:

- Gets notification about resource (application component) state changes during the online operation. This function runs after the agent initializes interface with the AMF kernel driver.

imf_register

This function performs the following task

- Registers the resource entities for online monitoring with the AMF kernel driver. The Apache agent reports the resource (application component) as online when the parent Apache HTTP server process and at least one child HTTP server process is running. The process ID (PID) of the parent Apache HTTP server process, and one child process found on the system, is registered with AMF. For example, the function registers the PID of the process that requires online

monitoring. This function runs for each resource after the resource goes into steady online state.

Chapter 2

Configuring application monitoring with Symantec ApplicationHA

This chapter includes the following topics:

About configuring application monitoring with ApplicationHA

This chapter describes the steps to configure application monitoring with ApplicationHA in a virtualization environment.

Consider the following points before you proceed:

- You configure an application for monitoring on a virtual machine using the Symantec ApplicationHA Configuration Wizard.
- The Symantec ApplicationHA Configuration Wizard is launched when you click **Configure Application Monitoring** in the Symantec High Availability view of the Veritas Operations Manager (VOM) Management Server console.
- In this release, the wizard allows you to configure monitoring for only one application per virtual machine.

To configure another application using the wizard, you must first unconfigure the existing application monitoring.

- After you have configured monitoring for an application using the wizard, you can configure monitoring for other applications residing in the same virtual machine, using Symantec Cluster Server (VCS) commands.
  For more information read the following technote:
  http://www.symantec.com/docs/TECH159846
- After configuring Apache HTTP Server for monitoring, if you create another Apache HTTP Server instance, this new instance is not monitored as part of the existing configuration.
  In such a case, you must first unconfigure the existing configuration and then reconfigure the application using the wizard. You can then select all the instances for monitoring.

Before configuring application monitoring for Apache HTTP Server

Ensure that you complete the following tasks before configuring application monitoring for Apache HTTP Server on a virtual machine:

- Install Veritas Operations Manager (VOM) Management Server. For more information on working with VOM, see the *Symantec ApplicationHA User's Guide*. For information on accessing the Symantec High Availability view: See "Accessing the Symantec High Availability view " on page 17.
- Install ApplicationHA guest components on the virtual machine that you need to monitor.
- Assign ApplicationHA - Configure Application Monitoring (Admin) privileges to the logged-on user on the virtual machine where you want to configure application monitoring.
- Install the application and the associated components that you wish to monitor on the virtual machine.
- If you have configured a firewall, ensure that your firewall settings allow access to ports used by ApplicationHA installer, wizards, and services.
  Refer to the *Symantec ApplicationHA Installation Guide* for a list of ports and services used.

Accessing the Symantec High Availability view

To administer an application on a virtual machine that is running in the KVM environment, you must access the Symantec High Availability view of the Veritas Operations Manager (VOM) Management Server console.

From the Symantec High Availability view, you can perform administrative actions such as:

- Start an application
- Stop an application
- Configure application monitoring
- Unconfigure application monitoring
- Enable application heartbeat
- Disable application heartbeat
- Enter maintenance mode
- Exit maintenance mode

To access the Symantec High Availability view

1. Log on to the VOM Management Server console.
2. Select the Server perspective and expand Manage in the left pane.
3. Expand the Organization, or Uncategorized Hosts to navigate to the virtual machine.
4. Right-click the required virtual machine, and then click **Manage ApplicationHA**.

   The Symantec High Availability view appears.

Configuring application monitoring for Apache HTTP Server

Perform the following steps to configure monitoring for Apache HTTP Server on a virtual machine.

To configure application monitoring for Apache HTTP Server

1. In the Symantec High Availability view of the VOM Management Server console, click **Configure Application Monitoring**.

   This launches the Symantec ApplicationHA Configuration Wizard.

2. Review the information on the Welcome screen and then click **Next**.

   The wizard lists all the supported applications for the system.

3. Select Apache, and then click **Next**.

   The Apache HTTP Server Instance Selection screen appears.

   Note: The wizard configures ApplicationHA to monitor Apache HTTP Server instances with Intelligent Monitoring Framework (IMF).

   ---

4. Enter the absolute path of the Apache HTTP Server executable file.
5. Enter the absolute path of the Apache HTTP Server configuration file.
6. Click **Add**.

   The Apache HTTP Server instance is added.

7. To add more Apache HTTP Server instances, repeat steps 2 to 6.
8. Click **Configure**.

   The wizard performs the application monitoring configuration tasks. The ApplicationHA Configuration screen displays the status of each task.

9. After all the tasks are complete, click **Next**.

   Note: If the configuration tasks fail, click **Diagnostic information** to check the details of the failure.

   After you check the details of the failure and resolve the issues, you have to run the wizard again to configure the application monitoring.

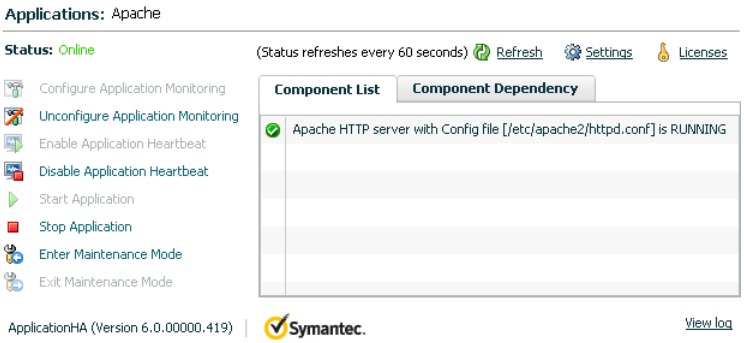
   ---

10. Click **Finish** to complete the wizard.

    This completes the application monitoring configuration.

11 To view the status of the configured application on a virtual machine, on the VOM Management Server console, right-click the appropriate virtual machine and then click **Manage ApplicationHA**.

The Symantec High Availability view appears.

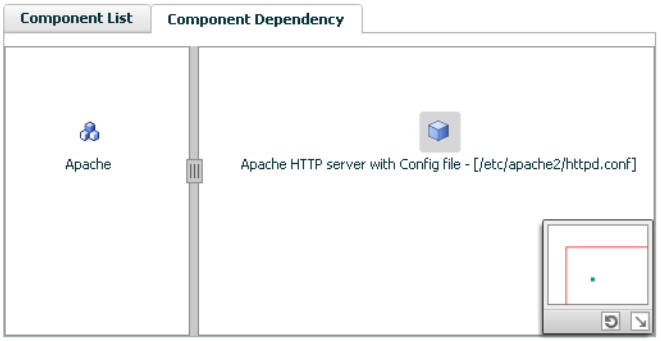


By default, the Component List tab appears. The tab lists each component of the configured application and the status description of each component.

For more information on viewing and administering applications through the Veritas Operations Manager, see the *Symantec ApplicationHA User's Guide*.

12 To view component dependency for the monitored application, click the **Component Dependency** tab.

The component dependency graph appears.



The graph illustrates the dependencies between a selected component group (an application or a group of inter-related components) and its components for the configured application. The left pane displays component groups and/or configured applications. The right pane displays components of the selected component group or application.

For more information on viewing component dependency for any configured application, see the *Symantec ApplicationHA User's Guide*.

Chapter 3

Troubleshooting the agent for Apache HTTP Server

This chapter includes the following topics:

Starting the Apache HTTP Server instance outside ApplicationHA control

If you face problems while working with an instance, you must disable the instance within the ApplicationHA framework. A disabled instance is not under the control of the ApplicationHA framework, and so you can test the Apache HTTP Server instance independent of the ApplicationHA framework. Refer to the *Symantec Cluster Server Administrator's Guide* for information about disabling a resource.

You can then restart the Apache HTTP Server instance outside the ApplicationHA framework.

Note: When you restart the instance outside the ApplicationHA framework, use the same parameters that the instance attributes define within the ApplicationHA framework.

When you bring an Apache HTTP Server online outside of ApplicationHA control, first source its environment file. Start the server with the `-f` option so the server knows which instance to start.

A sample procedure to start an Apache HTTP Server instance outside the ApplicationHA framework is illustrated as follows.

To restart the Apache HTTP Server outside the framework

1 Log in as an Apache User.

```
# su ApacheUser
```

2 Start the Apache HTTP Server.

```
$ httpdDir/envvars; Apache_HTTP_Server_executable -f ConfigFile
```

For example:

```
$ /apache/v2.2/bin/envars;/apache/v2.2/bin/httpd -f \
/apache/v2.2/conf/httpd.conf -k start
```

If the Apache HTTP Server works correctly outside the ApplicationHA framework, you can then attempt to implement the Apache HTTP Server within the ApplicationHA framework.

Monitoring Apache HTTP Server processes

The agent for Apache HTTP Server monitors all the processes similar to the following pattern:

Apache HTTP Server 2.0	`/usr/sbin/httpd -f \` `/etc/apache2/httpd.conf -k start`
IBM HTTP Server	`/opt/IBM/HTTPServer/bin/httpd -f \` `/opt/IBM/HTTPServer/conf/httpd.conf -k start`

Stopping Apache HTTP Server processes forcefully

When an attempt to gracefully stop the Apache HTTP Server fails, the agent for Apache HTTP Server kills all the processes similar to the following pattern:

Apache HTTP Server 2.0	`/usr/sbin/httpd -f \` `/etc/apache2/httpd.conf -k start`

IBM HTTP Server	`/opt/IBM/HTTPServer/bin/httpd -f \ /opt/IBM/HTTPServer/conf/httpd.conf -k start`

Reviewing error log files

If you face problems while using Apache HTTP Server or the agent for Apache HTTP Server, use the log files described in this section to investigate the problems.

Using Apache HTTP Server log files

If an Apache HTTP Server faces problems, you can access the server log files to diagnose the problem. Typically the Apache HTTP Server log files are located in the `/var/log/httpd` directory. Alternatively you can find the exact location of the log files, specified using the ErrorLog tag, in the Apache HTTP Server configuration file.

Reviewing ApplicationHA log files

If you face problems while using the agent for Apache HTTP Server, you can also access the ApplicationHA engine, Apache HTTP Server, and ApplicationHA log files for more information about a particular instance. The log files are located at the following location:

- The ApplicationHA engine log file is `/var/VRTSvcs/log/engine_A.log`
- Apache HTTP Server agent log file is `/var/VRTSvcs/log/Apache_A.log`
- ApplicationHA log file is `/var/VRTSvcs/log/AppControlOperations_A.log`

Using trace level logging

The LogDbg attribute controls the level of logging that is written in a log file for each Apache HTTP Server instance. You can set this attribute to DBG_5, which enables very detailed and verbose logging.

Reconfiguring ApplicationHA when Apache HTTP Server fails to start

This section describes the procedure to reconfigure application monitoring for Apache HTTP Server.

Perform the following steps to reconfigure application monitoring:

1. In the Symantec High Availability view of the VOM Management Server console, click **Unconfigure Application Monitoring**. A confirmation box appears.
2. Click **OK**.
3. Click **Configure** and proceed with configuring application monitoring for Apache HTTP Server. See "Configuring application monitoring for Apache HTTP Server" on page 17.

Appendix A

Resource type definitions

This appendix includes the following topics:

About the resource type and attribute definitions

The resource type represents the configuration definition of the agent and specifies how the agent is defined in the configuration file. The attribute definitions describe the attributes associated with the agent. The required attributes describe the attributes that must be configured for the agent to function.

Resource type definition for the Apache HTTP Server agent

The ApplicationHA agent for Apache HTTP Server is represented by the Apache HTTP Server resource type in ApplicationHA.

```
type Apache (
        static boolean IntentionalOffline = 0
        static keylist SupportedActions = { "checkconffile.vfd" }
        static str ArgList[] = { ResLogLevel, State, IState, httpdDir,
        SharedObjDir, EnvFile, PidFile, HostName, Port, User,
        SecondLevelMonitor, SecondLevelTimeout, ConfigFile, EnableSSL,
        DirectiveAfter, DirectiveBefore }
        static int IMF{} = { Mode = 2, MonitorFreq = 5,
        RegisterRetryLimit = 3 }
        static str IMFRegList[] = {ConfigFile,  httpDir }
```

```
        str ResLogLevel = INFO
        str httpdDir
        str SharedObjDir
        str EnvFile
        str PidFile
        str HostName
        int Port = 80
        str User
        boolean SecondLevelMonitor = 0
        int SecondLevelTimeout = 30
        str ConfigFile
        boolean EnableSSL = 0
        str DirectiveAfter{}
        str DirectiveBefore{}
)
```

Apache HTTP Server agent attributes

Table A-1 Required attributes

Required attributes	Description
ConfigFile	Specifies the full path and name of the main configuration file for the Apache HTTP Server. Type and dimension: string-scalar Example: "/apache/server1/conf/httpd.conf"
httpdDir	Full path of the Apache HTTP Server binary file or full path of the directory in which the httpd binary file is located. Type and dimension: string-scalar Example: "/apache/server1/bin"
ResLogLevel	This attribute has been deprecated. Use the resource type attribute LogDbg to enable debug logs. Set LogDbg attribute to DBG_5 to enable debug logs for the Apache HTTP server agent. By default, setting the LogDbg attribute to DBG_5 enables debug logs for all Apache resources in the cluster. If debug logs must be enabled for a specific Apache resource, override the LogDbg attribute. For information on how to use the LogDbg attribute, refer to the *Symantec Cluster Server Administrator's Guide*.

Table A-1 Required attributes *(continued)*

Required attributes	Description
EnvFile	This attribute may be required when you use IBM HTTP Server.

Table A-2 Optional attributes

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

Table A-2 Optional attributes *(continued)*

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

Table A-2 Optional attributes *(continued)*

Optional attributes	Description
SharedObjDir	Specifies the full path of the directory in which the Apache HTTP Server shared object files are located. This attribute is used when the HTTP Server is compiled using the SHARED_CORE rule. If you specify this attribute, the directory is passed to the `-R` option when executing the httpd program. Refer to the httpd man pages for more information about the `-R` option. Type and dimension: boolean-scalar Example: "/apache/server1/libexec"
SecondLevelMonitor	This attribute has been deprecated.
LevelTwoMonitorFreq	Specifies the frequency at which the agent must perform second-level or detailed monitoring. You can also override the value of this attribute at the resource level. The value indicates the number of monitor cycles after which the agent will monitor Apache in detail. For example, the value 5 indicates that the agent will monitor Apache in detail after every five online monitor intervals. Type and dimension: integer-scalar Default: 0
SecondLevelTimeout	Specifies the number of seconds that the monitor agent function waits on the execution of the second-level monitor. If the second-level monitor program does not return to calling the monitor agent function before the SecondLevelTimeout window expires, the monitor agent function no longer blocks on the program sub-process. It does, however, report that the resource is offline. The value should be high enough to allow the second level monitor enough time to complete. The value should be less than the value of the agent's MonitorTimeout. Type and dimension: integer-scalar Default: 30

Table A-3 shows the internal attributes for the agent for Apache HTTP Server.

Table A-3 Internal attribute for the agent for Apache

Internal attributes	Description
IMF	This resource-type level attribute determines whether the Apache HTTP Server agent must perform intelligent resource monitoring. This attribute includes the following keys: ■ Mode: Define this attribute to enable or disable intelligent resource monitoring. Valid values are as follows: ■ 0—Does not perform intelligent resource monitoring ■ 1—Performs intelligent resource monitoring for offline resources and performs poll-based monitoring for online resources ■ 2—Performs intelligent resource monitoring for online resources and performs poll-based monitoring for offline resources ■ 3—Performs intelligent resource monitoring for both online and for offline resources Default: 2 ■ MonitorFreq: This key value specifies the frequency at which the agent invokes the monitor agent function. The value of this key is an integer. Default: 5 You can set this key to a non-zero value for cases where the agent requires to perform both poll-based and intelligent resource monitoring. If the value is 0, the agent does not perform poll-based process check monitoring. After the resource registers with the AMF kernel driver, the agent calls the monitor agent function as follows: ■ After every (MonitorFreq x MonitorInterval) number of seconds for online resources ■ After every (MonitorFreq x OfflineMonitorInterval) number of seconds for offline resources ■ RegisterRetryLimit: If you enable intelligent resource monitoring, the agent invokes the imf_register agent function to register the resource with theAMF kernel driver. The value of the RegisterRetryLimit key determines the number of times the agent must retry registration for a resource. If the agent cannot register the resource within the limit that is specified, then intelligent monitoring is disabled until the resource state changes or the value of the Mode key changes. Default: 3

Appendix B

Detail monitoring

This appendix includes the following topics:

Setting the PATH variable

ApplicationHA commands reside in the `/opt/VRTS/bin` directory. Add this directory to your PATH environment variable.

To set the PATH variable

- Perform one of the following steps:

  For the Bourne Shell (sh or ksh), type:

  ```
  # PATH=/opt/VRTS/bin:$PATH; export PATH
  ```

  For the C Shell (csh or tcsh), type:

  ```
  # setenv PATH :/opt/VRTS/bin:$PATH
  ```

Setting up detail monitoring for a Apache HTTP Server instance

This section describes the procedure to enable and disable detail monitoring for Apache HTTP Server.

To enable detail monitoring for Apache HTTP Server

1. Make the ApplicationHA configuration writable:

   ```
   # haconf -makerw
   ```

2. Freeze the service group to avoid automated actions by ApplicationHA in case of an incomplete configuration:

   ```
   # hagrp -freeze Apache_1_SG
   ```

3. Enable detail monitoring for an Apache HTTP Server instance by using the following HA commands:

   ```
   # hares -override Apache_1_res LevelTwoMonitorFreq

   # hares -modify Apache_1_res LevelTwoMonitorFreq 1

   # hares -modify Apache_1_res HostName hostname

   # hares -modify Apache_1_res Port port_number
   ```

   Note: You can set the LevelTwoMonitorFreq attribute either at the resource type level or at the resource level. For more information about the LevelTwoMonitorFreq attribute, refer to the *Symantec Cluster Server Agent Developer's Guide*.

4. Unfreeze the service group:

   ```
   # hagrp -unfreeze Apache_1_SG
   ```

5. Make the ApplicationHA configuration read-only:

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

For example:

```
# haconf -makerw
# hagrp -freeze Apache_1_SG
# hares -override Apache_1_res LevelTwoMonitorFreq
# hares -modify Apache_1_res LevelTwoMonitorFreq 1
# hares -modify Apache_1_res HostName hostname
# hares -modify Apache_1_res Port port_number
# hagrp -unfreeze Apache_1_SG
# haconf -dump -makero
```

To disable detail monitoring for Apache HTTP Server

1. Make the ApplicationHA configuration writable:

   ```
   # haconf -makerw
   ```

2. Freeze the service group to avoid automated actions by ApplicationHA in case of an incomplete configuration:

   ```
   # hagrp -freeze Apache_1_SG
   ```

3. Disable detail monitoring for an Apache HTTP Server instance by using the following HA commands:

   ```
   # hares -modify Apache_1_res LevelTwoMonitorFreq 0
   ```

4. Unfreeze the service group.

   ```
   # hagrp -unfreeze Apache_1_SG
   ```

5. Make the ApplicationHA configuration read-only:

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