

Symantec™ ApplicationHA Agent for Apache HTTP Server Configuration Guide

Linux on KVM

6.0

Symantec™ ApplicationHA Agent for Apache HTTP Server Configuration 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.

Agent version: 6.0.000.000

Document version: 6.0.000.000.0

Legal Notice

Copyright © 2011 Symantec Corporation. All rights reserved.

Symantec, the Symantec 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. 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 Web site at the following URL:

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

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

Contacting Technical Support

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

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

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

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

- Product release level

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

Licensing and registration

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

www.symantec.com/business/support/

Customer service

Customer service information is available at the following URL:

www.symantec.com/business/support/

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

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

Support agreement resources

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

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

Documentation

Product guides are available on the 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 Web site.

<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

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

Technical Support	4	
Chapter 1	Introducing the Symantec ApplicationHA Agent for Apache HTTP Server	9
	About the Symantec agent for Apache HTTP Server	9
	About installing and removing the ApplicationHA agent for Apache HTTP Server	9
	Supported software	10
	Supported application versions	10
	Supported virtualization environments	10
	Supported operating systems on virtual machines	10
	Apache HTTP Server agent functions	11
	Online	11
	Offline	11
	Monitor	12
	Clean	12
Chapter 2	Configuring application monitoring with Symantec ApplicationHA	13
	About configuring application monitoring with ApplicationHA	13
	Before configuring application monitoring for Apache HTTP Server	14
	Accessing the ApplicationHA tab	15
	Configuring application monitoring for Apache HTTP Server	15
Chapter 3	Troubleshooting the agent for Apache HTTP Server	19
	Starting the Apache HTTP Server instance outside ApplicationHA control	19
	Monitoring Apache HTTP Server processes	20
	Stopping Apache HTTP Server processes forcefully	20
	Reviewing error log files	21
	Using Apache HTTP Server log files	21
	Reviewing ApplicationHA log files	21

	Using trace level logging	21
	Reconfiguring ApplicationHA when Apache HTTP Server fails to start	22
Appendix A	Resource type definitions	25
	About the resource type and attribute definitions	25
	Resource type definition for the Apache HTTP Server agent	25
	Apache HTTP Server agent attributes	26
Appendix B	Detail monitoring	31
	Setting the PATH variable	31
	Setting up detail monitoring for a Apache HTTP Server instance	31

Introducing the Symantec ApplicationHA Agent for Apache HTTP Server

This chapter includes the following topics:

- [About the Symantec agent for Apache HTTP Server](#)
- [About installing and removing the ApplicationHA agent for Apache HTTP Server](#)
- [Supported software](#)
- [Apache HTTP Server agent functions](#)

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.

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 6.0 installed.
- The following versions of the Veritas Operations Manager components are supported:
 - Veritas Operations Manager Management Server 4.1 or later
 - Veritas Operations Manager managed host for Linux: 4.1 or later
 - Veritas Operations Manager Add-on for Symantec ApplicationHA Management

Supported application versions

[Table 1-1](#) lists the Apache HTTP Server versions that Symantec ApplicationHA 6.0 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 1 in the physical host.

Supported operating systems on virtual machines

[Table 1-2](#) shows the supported operating systems for Symantec ApplicationHA 6.0.

Table 1-2 Supported guest operating systems

Operating systems	Levels	Kernel version
Red Hat Enterprise Linux 5	U3 or later	2.6.18-128.el5
Red Hat Enterprise Linux 6	Base or later	2.6.32-71.el6

Note: 64-bit operating systems are only 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 Veritas 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.

Configuring application monitoring with Symantec ApplicationHA

This chapter includes the following topics:

- [About configuring application monitoring with ApplicationHA](#)
- [Before configuring application monitoring for Apache HTTP Server](#)
- [Accessing the ApplicationHA tab](#)
- [Configuring application monitoring for Apache HTTP Server](#)

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 Application Monitoring Configuration Wizard.
- The Application Monitoring Configuration Wizard is launched when you click **Configure Application Monitoring** in the ApplicationHA tab of the Veritas Operations Manager (VOM) 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 Veritas 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 the Veritas Operations Manager (VOM) Management Server and install the VOM add-on for ApplicationHA on the VOM Management Server. For more information on working with VOM, see the *Symantec ApplicationHA User's Guide*. For information on Accessing the ApplicationHA tab: See "[Accessing the ApplicationHA tab](#)" on page 15.
- 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 ApplicationHA tab

To administer an application on a virtual machine that is running in the KVM environment, you must access the ApplicationHA tab of the Veritas Operations Manager (VOM) console.

In the ApplicationHA tab, 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 ApplicationHA tab

- 1 On the Veritas Operations Manager console, click **Manage > Servers > Hosts**.
- 2 In the left pane, in the **License** list box, select the **ApplicationHA** check box.
- 3 In the right pane, click the virtual machine where you want to perform administrative actions.
- 4 Click the **ApplicationHA** tab.

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 ApplicationHA tab of the Veritas Operations Manager console, click **Configure Application Monitoring**.

This launches the Application Monitoring 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.

- 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 **3** 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 **View Logs** to check the details of the failure.

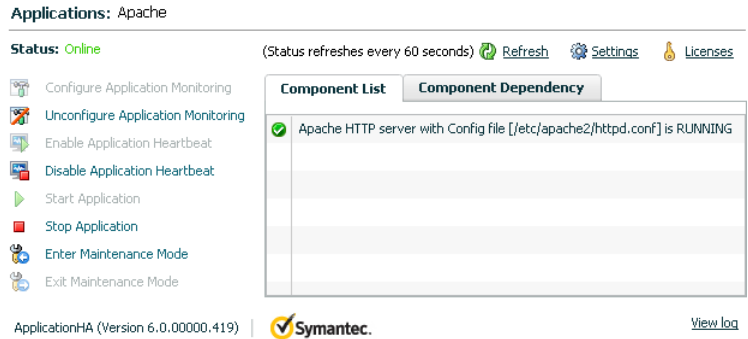
After you check the details of the failure and resolve the issues, you have to run the wizard again to configure the application monitoring.

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

This completes the application monitoring configuration.

- To view the status of the configured application on a virtual machine, on the Veritas Operations Manager console, navigate to the appropriate virtual machine, and then click the **ApplicationHA** tab.

The ApplicationHA view appears.

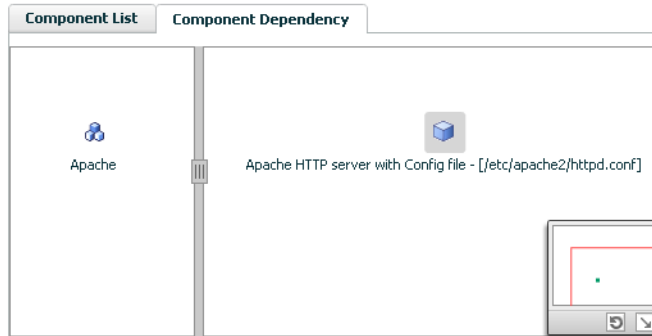


By default, the Component List tab appears. The tab lists each component of the configured application and the status description of each component.

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*.

Troubleshooting the agent for Apache HTTP Server

This chapter includes the following topics:

- [Starting the Apache HTTP Server instance outside ApplicationHA control](#)
- [Monitoring Apache HTTP Server processes](#)
- [Stopping Apache HTTP Server processes forcefully](#)
- [Reviewing error log files](#)
- [Reconfiguring ApplicationHA when Apache HTTP Server fails to start](#)

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 *Veritas 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/envvars;/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 /usr/sbin/httpd -f \  
Server 2.0 /etc/apache2/httpd.conf -k start
```

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

```
IBM HTTP      /opt/IBM/HTTPServer/bin/httpd -f \
Server        /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 `ResLogLevel` 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 `TRACE`, which enables very detailed and verbose logging.

If you set `ResLogLevel` to `TRACE`, a very high volume of messages is produced. Symantec recommends that you localize the `ResLogLevel` attribute for a particular instance.

To localize ResLogLevel attribute for a resource

- 1 ApplicationHA commands reside in the `/opt/VRTS/bin` directory. Add this directory to your PATH environment variable. To set the path variable, perform the following step:

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

- 2 Make the ApplicationHA configuration writable:

```
# haconf -makerw
```

- 3 Identify the instance for which you want to enable detailed logging.

- 4 Localize the ResLogLevel attribute for the identified resource:

```
# hares -local Resource_Name ResLogLevel
```

- 5 Set the ResLogLevel attribute to TRACE for the identified resource:

```
# hares -modify Resource_Name ResLogLevel TRACE -sys SysA
```

- 6 Note the time before you begin to operate the identified resource.

- 7 Test the identified resource. The function reproduces the problem that you are attempting to diagnose.

- 8 Note the time when the problem is reproduced.

- 9 Set the ResLogLevel attribute back to INFO for the identified resource:

```
# hares -modify Resource_Name ResLogLevel INFO -sys SysA
```

- 10 Save the configuration.

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

- 11 Review the contents of the log file. Use the time noted in Step 6 and Step 8 to diagnose the problem.

You can also contact Symantec support for more help.

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 ApplicationHA tab of the Veritas Operations Manager 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 15.

Resource type definitions

This appendix includes the following topics:

- [About the resource type and attribute definitions](#)
- [Resource type definition for the Apache HTTP Server agent](#)
- [Apache HTTP Server agent attributes](#)

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 }
    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	<p>Specifies the full path and name of the main configuration file for the Apache HTTP Server.</p> <p>Type and dimension: string-scalar</p> <p>Example: "/apache/server1/conf/httpd.conf"</p>
httpdDir	<p>Specifies the full path of directory to the the Apache HTTP Server executable.</p> <p>Type and dimension: string-scalar</p> <p>Example: "/apache/server1/bin"</p>
ResLogLevel	<p>Controls the agent's logging detail for a specific instance of a resource. The values are:</p> <ul style="list-style-type: none"> ■ ERROR: Logs error messages. ■ WARN: Logs error and warning messages. ■ INFO: Logs error, warning, and informational messages. ■ TRACE: Logs error, warning, informational, and trace messages. Trace logging is verbose. Use it for initial configuration or troubleshooting. <p>Type and dimension: boolean-scalar</p> <p>Default: 0</p> <p>Example: "TRACE"</p>

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

Table A-2 Optional attributes (*continued*)

Optional attributes	Description
<p>HostName</p>	<p>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.</p> <p>You can use IPv4 or IPv6 addresses for the HostName attribute.</p> <p>Note: The HostName attribute is only required when the value of SecondLevelMonitor is 1 (true).</p> <p>Type and dimension: string-scalar</p> <p>Example: "web1.example.com"</p>
<p>Port</p>	<p>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 SecondLevelMonitor is set to 1 (true).</p> <p>Type and dimension: integer-scalar</p> <p>Default: 80</p> <p>Example: "80"</p>
<p>EnvFile</p>	<p>Specifies the path and name of the file that is sourced before executing the <code>httpdDir/httpd</code> program. With Apache 2.0, the file <code>ServerRoot/bin/envvars</code>, which is supplied in most Apache 2.0 distributions, is commonly used to set the environment before executing <code>httpd</code>. Specifying this attribute is optional. If <code>EnvFile</code> is specified, the shell for user must be Bourne, Korn, or C shell.</p> <p>This attribute may be required when you use the IBM HTTP Server if the online action fails. For example: Set the <code>EnvFile</code> to <code>/usr/IBM/HTTPServer/bin/envvars</code>.</p> <p>Type and dimension: string-scalar</p> <p>Example: <code>"/apache/server1/bin/envvars"</code></p>
<p>PidFile</p>	<p>This attribute is for internal-use only.</p>

Table A-2 Optional attributes (*continued*)

Optional attributes	Description
SharedObjDir	<p>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 <code>-R</code> option when executing the <code>httpd</code> program. Refer to the <code>httpd</code> man pages for more information about the <code>-R</code> option.</p> <p>Type and dimension: boolean-scalar</p> <p>Example: <code>"/apache/server1/libexec"</code></p>
SecondLevelMonitor	<p>Enables second-level monitoring for the resource. Second-level monitoring is a deeper, more thorough state check of the Apache HTTP Server. The valid attribute values are 1 (true) and 0 (false).</p> <p>Type and dimension: boolean-scalar</p> <p>Default: 0</p> <p>Example: <code>"1"</code></p>
SecondLevelTimeout	<p>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 <code>SecondLevelTimeout</code> 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 <code>MonitorTimeout</code>.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 30</p>

Detail monitoring

This appendix includes the following topics:

- [Setting the PATH variable](#)
- [Setting up detail monitoring for a Apache HTTP Server instance](#)

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:

```
# hagrps -freeze Apache_1_SG
```

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

```
# hares -modify Apache_1_res SecondLevelMonitor frequency
```

```
# hares -modify Apache_1_res HostName hostname
```

```
# hares -modify Apache_1_res Port port_number
```

Note: For more information on SecondLevelMonitor attribute, See [“Apache HTTP Server agent attributes”](#) on page 26.

- 4 Unfreeze the service group:

```
# hagrps -unfreeze Apache_1_SG
```

- 5 Make the ApplicationHA configuration read-only:

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

For example:

```
# haconf -makerw
# hagrps -freeze Apache_1_SG
# hares -modify Apache_1_res SecondLevelMonitor frequency
# hares -modify Apache_1_res HostName hostname
# hares -modify Apache_1_res Port port_number
# hagrps -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:

```
# hagrps -freeze Apache_1_SG
```


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

```
# hares -modify Apache_1_res SecondLevelMonitor 0
```

- 4 Unfreeze the service group.

```
# hagrps -unfreeze Apache_1_SG
```

- 5 Make the ApplicationHA configuration read-only:

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

