

Symantec™ ApplicationHA Agent for WebSphere Application Server Configuration Guide

Linux on VMware

6.0

Symantec™ ApplicationHA Agent for WebSphere Application 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: 5.1.8.0

Document version: 5.1.8.0.2

Legal Notice

Copyright © 2012 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 WebSphere Application Server agent	9
About the Symantec agent for WebSphere Application Server	9
About installing and removing the ApplicationHA agent for WebSphere Application Server	10
Supported software	10
Supported application versions	10
Supported VMware versions	10
Supported guest operating systems	11
WebSphere Application Server agent functions	12
Online	12
Offline	12
Monitor	13
Clean	14
Chapter 2 Configuring application monitoring with ApplicationHA	15
About configuring application monitoring with ApplicationHA	15
Before configuring application monitoring for WebSphere Application Server	16
Configuring application monitoring for WebSphere Application Server	17
Chapter 3 Troubleshooting the WebSphere Application Server agent	21
Starting the WebSphere Application Server instance outside ApplicationHA control	21
Reviewing error log files	23
Reviewing cluster log files	23
Reviewing agent log files	24
Using trace level logging	24

	Defining additional environment variables for a WebSphere Application Server instance	25
Appendix A	Resource Type Definitions	27
	About the resource type and attribute definitions	27
	Resource type definition for WebSphere Application Server	27
	WebSphere Application Server agent attributes	28
Appendix B	Detail Monitoring	33
	Setting the PATH variable	33
	Setting up detail monitoring for ApplicationHA agent for WebSphere Application Server	33

Introducing the WebSphere Application Server agent

This chapter includes the following topics:

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

About the Symantec agent for WebSphere Application Server

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

The Symantec agent for WebSphere Application Server provides high availability for WebSphere Application Server in a virtual machine.

The agent supports the following types of WebSphere Application Server instances:

- Deployment Manager
- Node Agent
- Application Server

About installing and removing the ApplicationHA agent for WebSphere Application Server

When you install or uninstall Symantec ApplicationHA, the ApplicationHA agent for WebSphere Application 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 WebSphere Application Server is automatically installed or removed. For more information, see the *Symantec ApplicationHA Agent Pack Installation Guide*.

Supported software

The Symantec ApplicationHA agent for WebSphere Application Server supports the following software versions:

- Symantec ApplicationHA agent for WebSphere Application 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 WebSphere Application Server versions that Symantec ApplicationHA 6.0 currently supports on virtual machine.

Table 1-1 Supported application versions

Application	Version
WebSphere Application Server	7.x and 8.0

Supported VMware versions

The following VMware Servers and management clients are currently supported:

- VMware ESX Server version 4.0 (for ApplicationHA initiated reboot only), 4.1, 4.1 Update 1
- VMware ESXi Server version 4.0, 4.1, 5.0
- VMware vCenter Server version 4.0, 4.1, 4.1 Update 1, 5.0

Note: VMware Fault Tolerance is not supported in case of vCenter Server 4.1

- VMware vSphere Client version 4.0, 4.1, 5.0

Supported guest operating systems

[Table 1-2](#) shows the supported operating systems for this release.

Table 1-2 Supported guest operating systems

Operating systems	Levels	Kernel version
Oracle Enterprise Linux 5	U3 or later	2.6.18-128.0.0.0.1.el5
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
SUSE Linux Enterprise 10	SP4 or later	2.6.16.60-0.75.1
SUSE Linux Enterprise 11	Base or later	2.6.32.12-0.7-default

Note: 64-bit operating systems are only supported.

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

Symantec supports only Oracle, Red Hat, and SUSE distributed kernel binaries.

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

WebSphere Application 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

The online function is responsible for starting a WebSphere Application Server instance. The online function performs the following tasks:

- Verifies that the WebSphere Application Server instance is not already online.
- Determines the version of the WebSphere Application Server software.
- Starts the WebSphere Application Server instance by executing the appropriate start script, which is supplied by the WebSphere Application Server installation program. The script executed depends upon the type of server being started.

Server Type	Start Command
Deployment Manager	<code>WAS_HOME/binDir/startManager.sh</code>
Node Agent	<code>WAS_HOME/binDir/startNode.sh</code>
Application Server	<code>WAS_HOME/binDir/startServer.sh</code>

Note: WAS_HOME is the location where WebSphere Application Server is installed.

Offline

The offline function is responsible for stopping a WebSphere Application Server instance. The offline function performs the following tasks:

- Verifies that the WebSphere Application Server instance is not already offline.
- Determines the version of the WebSphere Application Server.
- Stops the WebSphere Application Server instance by executing the appropriate stop script, which is supplied by the WebSphere Application Server installation program. The script executed depends upon the type of server being stopped.

Server Type	Stop Command
Deployment Manager	<code>WAS_HOME/binDir/stopManager.sh</code>
Node Agent	<code>WAS_HOME/binDir/stopNode.sh</code>

Application Server WAS_HOME/binDir/stopServer.sh

Monitor

The monitor function is responsible for monitoring the state of WebSphere Application Server instances on all nodes in the cluster.

The monitor function performs the following tasks:

- First-level monitoring quickly checks for the existence of the system process (the Java virtual machine) that represents the WebSphere Application Server instance. It determines the process existence by scanning the system process table and searching for strings in the process command line that uniquely identify the JVM process associated with the WebSphere Application Server instance. These search strings include the values specified in agent attributes WAS_HOME, WAS_NODE, and ServerName.

- If second-level monitoring is enabled (if SecondLevelMonitor > 0), the monitor function performs a deeper, more thorough state check of WebSphere Application Server. Second-level monitoring uses the IBM-supplied utility program serverStatus.sh. The output from this program is parsed to confirm the server is running.

When enabled, the integer value specified in attribute SecondLevelMonitor determines how frequently the program is executed. For example, if SecondLevelMonitor is set to 1, the monitor function executes serverStatus.sh during each monitor interval. If SecondLevelMonitor is set to 3, the monitor function executes serverStatus.sh every third monitor interval. This mechanism lets you control the system load generated by monitoring.

The serverStatus.sh script spawns a Java program that establishes a connection to WebSphere Application Server. Spawning a JVM every monitor interval places additional load on the system. If performance is more important than a second-level state check, then consider disabling second-level monitoring and only performing the first-level process check.

- The monitor function executes a custom monitor program specified in the attribute MonitorProgram. This program does not execute if either the first- or second-level monitor reports that the instance is offline. You can omit second-level monitoring, and attempt running a custom monitor check immediately after first-level monitoring.

Note: You cannot configure second-level or detail monitoring by using the ApplicationHA wizard. To enable detail monitoring, see Appendix B (Detail monitoring) of this guide.

This feature allows ApplicationHA administrator to define custom programs that determine the state of WebSphere Application Server. For example, the administrator may want to test the status of a J2EE component running inside the server and ensure that the underlying application is functioning properly. See [“WebSphere Application Server agent attributes”](#) on page 28.

Clean

The clean function removes any WebSphere Application Server instance processes remaining after a fault event or after an unsuccessful attempt to online or offline the instance.

The clean function performs the following tasks:

- Kills the process that starts the WebSphere Application Server instance. It is unlikely that this process exists, but it needs to be removed if for some reason it still exists during clean.
- Kills the process that stops the WebSphere Application Server instance. It is unlikely this process exists, but it needs to be removed if for some reason it still exists during clean.
- Kills the JVM process for the WebSphere Application Server instance. This process is identified by searching the system process table using the values specified in attributes WAS_HOME, WAS_NODE, and ServerName.

Configuring application monitoring with ApplicationHA

This chapter includes the following topics:

- [About configuring application monitoring with ApplicationHA](#)
- [Before configuring application monitoring for WebSphere Application Server](#)
- [Configuring application monitoring for WebSphere Application 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** on the ApplicationHA tab of the VMware vSphere Client.
- 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 WebSphere Application Server for monitoring, if you create another WebSphere Application 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 WebSphere Application Server

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

- Install ApplicationHA Console.
- Install ApplicationHA guest components on the virtual machine that you need to monitor.
- Install VMware Tools on the virtual machine. Install a version that is compatible with VMware ESX server.
- Install the VMware vSphere Client.
- 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 and Upgrade Guide* for a list of ports and services used.

Configuring application monitoring for WebSphere Application Server

Perform the following steps to configure monitoring for WebSphere Application Server on a virtual machine.

To configure application monitoring for WebSphere Application Server

- 1 Launch the VMware vSphere Client and connect to the VMware vCenter Server that hosts the virtual machine.

The vSphere Client is used to configure and control application monitoring.
- 2 From the vSphere Client's Inventory view in the left pane, select the virtual machine where you want to configure application monitoring for WebSphere Application Server.
- 3 From the vSphere Client's Management view in the right pane, click the **ApplicationHA** tab.

The ApplicationHA view displays the status of all the supported applications that are installed on the selected virtual machine.
- 4 In the ApplicationHA view, click **Configure Application Monitoring**.

This launches the Application Monitoring Configuration Wizard.
- 5 Review the information on the Welcome screen and then click **Next**.

The wizard lists all the supported applications for the system.
- 6 Select WebSphere Application Server, and then click **Next**.
- 7 Enter the WAS HOME directory name and then click **Add**.
- 8 Review the selected WAS HOME and then click **Next**.

The WebSphere Application Server Instance Selection screen appears.

Note: The wizard may take some time to discover the WebSphere Application Server instances. The response time depends on the execution of WebSphere-supplied commands.

- 9 Select the WebSphere Application Server instances that you want to monitor and then click **Configure**.

- 10** The wizard performs the application monitoring configuration tasks. The ApplicationHA Configuration screen displays the status of each task. After all the tasks are complete, click **Next**.

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

You then have to run the wizard again to configure the application monitoring.

- 11** Click **Finish** to complete the wizard.
This completes the application monitoring configuration.

- 12
- To view the status of the configured application on a virtual machine, in the inventory view of the vSphere Client, click the appropriate virtual machine, and then click the **ApplicationHA** tab.

The ApplicationHA view appears.

Applications: WebSphere Application Server

Status: Online (Status refreshes every 60 seconds) Refresh Settings Licenses

Configure Application Monitoring

Unconfigure Application Monitoring

Enable Application Heartbeat

Disable Application Heartbeat

Start Application

Stop Application

Enter Maintenance Mode

Exit Maintenance Mode

Component List

Component Dependency

	WebSphere Application Server [nodeagent : NodeAgent] inside Node [AppSer01Node] is RUNNING
	WebSphere Application Server [server1 : ApplicationServer] inside Node [AppSer01Node] is RUNNING
	WebSphere Application Server [dmgr : DeploymentManager] inside Node [Dmgr001Node] is RUNNING

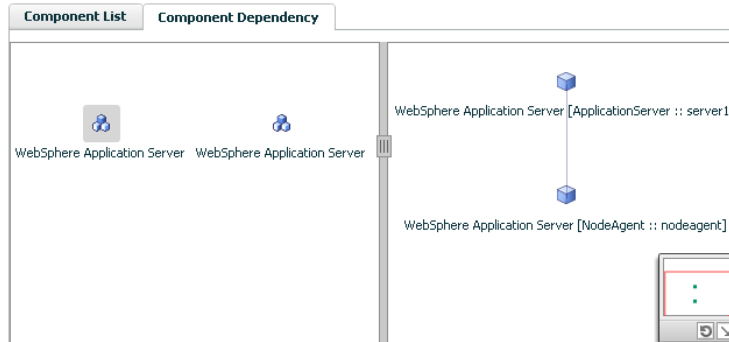
ApplicationHA (Version 6.0.00000.397) | powered by: Symantec. View log

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 by using the vSphere Client, see the Symantec *ApplicationHA User’s Guide*.

- 13** 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 WebSphere Application Server agent

This chapter includes the following topics:

- [Starting the WebSphere Application Server instance outside ApplicationHA control](#)
- [Reviewing error log files](#)
- [Defining additional environment variables for a WebSphere Application Server instance](#)

Starting the WebSphere Application 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 WebSphere Application Server instance independent of the ApplicationHA framework.

You can then restart the WebSphere Application Server instance outside the ApplicationHA framework.

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

A sample procedure to start a WebSphere Application Server instance outside the ApplicationHA framework, is illustrated as follows.

To start a WebSphere Deployment Manager outside the ApplicationHA framework

- 1
- Using the user name specified in the User attribute, log into the host on which the WebSphere Deployment Manager application is to run.
- 2
- Use the values specified in the agent attributes to start the WebSphere Deployment Manager.

For example, assume that the WebSphere Deployment Manager environment is set as follows:

Attribute	Value
ServerType	DeploymentManager
ServerName	dmgr
WAS_NODE	was70c1dmsol
WAS_HOME	/ibm/was/v70
ServerProfile	Dmgr01
	Specify this attribute for WebSphere version 6.0 and later

- 3
- Go to specified directory.
`/ibm/was/v70/profiles/Dmgr01/bin`
- 4
- Using the startManager.sh script, start the Deployment Manager.
`/ibm/was/v70/profiles/Dmgr01/bin/startManager.sh`
- 5
- Ensure that the Deployment Manager Server starts successfully.
If the Deployment Manager works properly outside the ApplicationHA framework, you can attempt to implement the server within the framework.

To start a WebSphere node agent outside the ApplicationHA framework

- 1 Using the user name specified in the User attribute, log into the host on which the WebSphere Node agent application is to run.
- 2 Use the values specified in the agent attributes to start the WebSphere Node agent.

For example, assume that the WebSphere Node agent environment is set as follows:

Attribute	Value
ServerType	NodeAgent
ServerName	nodeagent
WAS_NODE	was70c1n1sol
WAS_HOME	/ibm/was/v70
ServerProfile	AppSrv01

- 3 Go to specified directory.

```
/ibm/was/v70/profiles/AppSrv01/bin
```

- 4 Using the startNode.sh script, start the Node Agent:

```
/ibm/was/v70/profiles/AppSrv01/bin/startNode.sh
```

- 5 Ensure that the Node Agent starts successfully.

If the Node Agent works properly outside the ApplicationHA framework, you can attempt to implement the server within the framework.

Reviewing error log files

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

Reviewing cluster log files

In case of problems while using the agent for WebSphere Application Server, you can also access the engine log file for more information about a particular instance. The engine log files are located at the following location:

- The VCS engine log file is `/var/VRTSvcS/log/engine_A.log`

- WebSphere Application Server agent log file is
`/var/VRTSvcs/log/WebSphere_A.log`
- ApplicationHA log file is `/var/VRTSvcs/log/AppControlOperations_A.log`

Reviewing agent log files

In case of problems while using the agent for WebSphere, you can access the agent log files for more information. The agent saves output of every entry point process in the temporary folder of the resource system. If the temporary folder is `/tmp`, the log files are saved using the following naming format:

```
/tmp/.VRTSAgentName/ResourceName_EntryPointName.Process_ID
```

For example, for a resource `WAS50DeployMgr_dmgr`:

```
/tmp/.VRTSWebSphere/WAS50DeployMgr_dmgr.online.Process_ID  
/tmp/.VRTSWebSphere/WAS50DeployMgr_dmgr.offline.Process_ID  
/tmp/.VRTSWebSphere/WAS50DeployMgr_dmgr.clean.Process_ID  
/tmp/.VRTSWebSphere/WAS50DeployMgr_dmgr.monitor.Process_ID
```

If a resource, `WAS50DeployMgr_dmgr` is unable to bring a WebSphere Node Manager online, you can access the `/tmp/.VRTSWebSphere/WAS50DeployMgr_dmgr.online.Process_ID` for more information so that you can diagnose the problem.

Note: These files are overwritten each time you execute the corresponding agent function process. In case you want to save the information, make a copy of the files at another location.

Using trace level logging

The `ResLogLevel` attribute controls the level of logging that is written in a cluster log file for each WebSphere Application Server resource. 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 are produced. Symantec recommends that you localize the `ResLogLevel` attribute for a particular instance.

To localize `ResLogLevel` attribute for an instance

- 1 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`
- 2 Identify the resource for which you want to enable detailed logging.
 - 3 Localize the ResLogLevel attribute for the identified resource:

```
# hares -local Resource_Name ResLogLevel
```
 - 4 Set the ResLogLevel attribute to TRACE for the identified resource:

```
# hares -modify Resource_Name ResLogLevel TRACE -sys SysA
```
 - 5 Note the time before you begin to operate the identified resource.
 - 6 Test the identified resource. The function reproduces the problem that you are attempting to diagnose.
 - 7 Note the time when the problem is reproduced.
 - 8 Set the ResLogLevel attribute back to INFO for the identified resource:

```
# hares -modify Resource_Name ResLogLevel INFO -sys SysA
```
 - 9 Review the contents of the log file. Use the time noted in Step 4 and Step 6 to diagnose the problem.
- You can also contact Symantec support for more help.

Defining additional environment variables for a WebSphere Application Server instance

The WebSphere Application Server uses the `setupCmdLine.sh` file, which defines the required environment needed for WebSphere Application Server. The file is sourced in all the start, stop, and monitoring scripts used by the WebSphere Application Server agent.

```
bash-3.00# cat
/opt/IBM/WebSphere/AppServer/profiles/AppSrv01/bin/startServer.sh

#!/bin/sh

binDir=`dirname ${0}`

. ${binDir}/setupCmdLine.sh
```

```
${WAS_HOME}/bin/startServer.sh "$@"
```

By design, the WebSphere Application Server agent does not source the user's profile. If you need to export any additional environment variables, create an environment file with the required variables and source it in following scripts:

```
startManager.sh, startNode.sh, startServer.sh, stopManager.sh,  
stopNode.sh, stopServer.sh, serverStatus.sh
```

For example, append a line in `startServer.sh` in the following manner:

```
# Call User Environment here.  
. <Path to env file>/setUserEnv.sh
```

Resource Type Definitions

This appendix includes the following topics:

- [About the resource type and attribute definitions](#)
- [WebSphere Application 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 WebSphere Application Server

The ApplicationHA agent for WebSphere Application Server is represented by the WebSphere resource type in ApplicationHA.

```
type WebSphere (
    static boolean AEPTIMEOUT = 1
    static str AgentFile = "/opt/VRTSvcs/bin/Script50Agent"
    static str AgentDirectory = "/opt/VRTSagents/ha/bin/WebSphere"
    static str ArgList [] = { ResLogLevel, State, IState,
        ServerName, WAS_NODE,
        WAS_HOME, User, ServerProfile, ServerType, StartOptions,
        StopOptions, MonitorProgram, SecondLevelMonitor }
    str ResLogLevel = "INFO"
    str ServerName
    str WAS_NODE
    str WAS_HOME
    str User
```

```
        str  ServerProfile
        str  ServerType
        str  StartOptions
        str  StopOptions
        str  MonitorProgram
        int  SecondLevelMonitor = 0
    )
```

WebSphere Application Server agent attributes

Table A-1 shows the required attributes for the agent for WebSphere Application Server.

Table A-1 Required attributes

Required attribute	Description
ResLogLevel	<p>The logging detail performed by the agent for the resource. Valid values are:</p> <p>ERROR: Only logs error messages.</p> <p>WARN: Logs above plus warning messages.</p> <p>INFO: Logs above plus informational messages.</p> <p>TRACE: Logs above plus trace messages. TRACE is very verbose and should only be used during initial configuration or for troubleshooting and diagnostic functions.</p> <p>Type and dimension: string-scalar</p> <p>Default: INFO</p> <p>Example: TRACE</p>
SecondLevelMonitor	<p>Specifies if second-level monitor is enabled and how frequently it is performed. Second-level monitor is a deeper, more thorough state check of the WebSphere resource, performed by executing the IBM-supplied utility program serverStatus.sh. The output from this program is parsed to confirm the server status is running. The integer value specified by this attribute determines how frequently the second-level monitor program is executed.</p> <p>For example, if SecondLevelMonitor is set to 1, the monitor function will execute serverStatus.sh during each monitor interval. A value of 3 executes the program every third monitor interval. If SecondLevelMonitor is set to 0, the monitor function will never perform the second-level monitor.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 0</p> <p>Example: 1</p>

Table A-1 Required attributes (*continued*)

Required attribute	Description
ServerName	<p>Contains the server name assigned to the WebSphere Application Server during its installation. In Network Deployment configurations, the default ServerName for Deployment Managers is dmgr and the default ServerName for the Node Agents is nodeagent, but these names are not mandatory.</p> <p>Type and dimension: string-scalar</p> <p>Default: ""</p> <p>Example: server1</p>
ServerProfile	<p>Server profile name of the WebSphere Application Server instance or complete path to the WebSphere Application Server profile.</p> <p>Type and dimension: string-scalar</p> <p>Default: ""</p> <p>Example 1: Dmgr01</p> <p>Example 2: /WAS/AppSrv/profiles/AppSrv01/</p>
ServerType	<p>Type of WebSphere Application Server that the cluster will manage. Valid names are as follows:</p> <ul style="list-style-type: none"> ■ DeploymentManager: Instance is a Deployment Manager. ■ NodeAgent: Instance is a Node Agent. ■ ApplicationServer: Instance is an Application Server, which may be a stand-alone server or may be part of a Network Deployment and is a member of a WebSphere Cell. <p>The agent uses this value to determine how to manage the WebSphere Application Server within a cluster. Refer to the WebSphere documentation for a full explanation of the purposes and use of each WebSphere Application Server type.</p> <p>Type and dimension: string-scalar</p> <p>Default: ""</p> <p>Example: DeploymentManager</p>
User	<p>The UNIX user name used to run the programs that start, stop, and monitor the WebSphere instance, which include the program specified in the MonitorProgram attribute. IBM recommends using the root account, but you may use any account. If User is not set to root, the user name must be synchronized across the systems within the cluster. In other words, the user name must resolve to the same UID and have the same default shell on each system in the cluster.</p> <p>Type and dimension: string-scalar</p> <p>Default: ""</p> <p>Example: root</p>

Table A-1 Required attributes (continued)

Required attribute	Description
WAS_HOME	<p>The absolute path to the WebSphere Application Server or WebSphere Application Server Network Deployment root installation directory. This attribute is used to locate programs executed by the agent. It is also where the <i>binDir/setupCmdLine.sh</i> file resides. The value is also used to uniquely identify the ServerType processes. Using WAS_HOME to uniquely identify an Application Server's process IDs requires that WAS_HOME be unique compared to WAS_HOME for all other WAS instances in the cluster.</p> <p>Note: Both WAS_HOME and WAS_ND_HOME are defined as WAS_HOME in the standard environment file <i>setupCmdLine.sh</i>, which is supplied with WebSphere.</p> <p>Type and dimension: string-scalar</p> <p>Default: ""</p> <p>Example: /ibm/was</p>
WAS_NODE	<p>The WebSphere Node Name to which the server instance belongs. The Node Name is an administrative identifier that is internal to the WebSphere environment and is assigned when the node is installed. WebSphere requires that a Node Name must be unique within a WebSphere cell.</p> <p>Type and dimension: string-scalar</p> <p>Default: ""</p> <p>Example: was51c1n2</p>

Table A-2 lists the optional attributes for the agent for WebSphere Application Server.

Table A-2 Optional attributes

Optional Attribute	Definition
MonitorProgram	<p>The full pathname and command-line arguments for an externally-provided custom monitor program. The program is executed within the security context of the UNIX account specified in attribute User. The program must be completely self-contained and independent, and it must return one of the following exit codes:</p> <p>110 or 0: The WebSphere Application Server is ONLINE.</p> <p>100 or 1: The WebSphere Application Server is OFFLINE.</p> <p>All other: The WebSphere Application Server state is UNKNOWN.</p> <p>Symantec recommends storing the external monitor program on the shared storage device, in the directory specified by the WAS_HOME attribute, to ensure the file is always available on the online system.</p> <p>Type and dimension: string-scalar</p> <p>Default: ""</p> <p>Example: /usr/WAS/server1/bin/mymonitor.sh</p>
StartOptions	<p>The command-line options that are passed to the WebSphere start script when it is executed within the online function. Multiple options should be separated by a space. Refer to the WebSphere product documentation for a list and description of supported start options.</p> <p>Type and dimension: string-scalar</p> <p>Default: ""</p> <p>Example: "-replacelog -trace"</p>
StopOptions	<p>The command-line options that are passed to the WebSphere stop script when it is executed within the offline function. Multiple options should be separated by a space. Refer to the WebSphere product documentation for a list and description of supported stop options.</p> <p>Type and dimension: string-scalar</p> <p>Default: ""</p> <p>Example: "-replacelog -trace"</p>

Detail Monitoring

This appendix includes the following topics:

- [Setting the PATH variable](#)
- [Setting up detail monitoring for ApplicationHA agent for WebSphere Application Server](#)

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 ApplicationHA agent for WebSphere Application Server

This section describes the procedure to enable and disable detail monitoring for WebSphere Application Server.

To enable detail monitoring for WebSphere Application 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 WAS_<CellName>_<NodeName>_SG
```

- 3 Enable detail monitoring for WebSphere Application Server instances by using the following ApplicationHA commands:

```
# hares -modify WAS_<CellName>_<NodeName>_<ServerName>_res  
SecondLevelMonitor <frequency>
```

Note: For more information on SecondLevelMonitor attribute, See [“WebSphere Application Server agent attributes”](#) on page 28.

- 4 Save the configuration and unfreeze the service group.

```
# hagrps -unfreeze WAS_<CellName>_<NodeName>_SG  
  
# haconf -dump -makero
```

To disable detail monitoring for WebSphere Application 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 WAS_<CellName>_<NodeName>_SG
```

- 3 Disable detail monitoring for ApplicationHA instances by using the following ApplicationHA commands:

```
# hares -modify WAS_<CellName>_<NodeName>_<ServerName>_res  
SecondLevelMonitor 0
```

- 4 Save the configuration and unfreeze the service group.

```
# hagrps -unfreeze WAS_<CellName>_<NodeName>_SG  
  
# haconf -dump -makero
```