Symantec™ ApplicationHA Agent for WebLogic Server Installation and Configuration Guide

Linux

5.1

Symantec™ ApplicationHA Agent for WebLogic Server Installation and Configuration Guide

Product version: 5.1

Document version: 5.1.1

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 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

Documentation

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

docs@symantec.com

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

Contents

Chapter 1

Introducing the Symantec ApplicationHA Agent for WebLogic Server

This chapter includes the following topics:

About the Symantec ApplicationHA agent for WebLogic Server

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

The Symantec ApplicationHA agent for WebLogic Server provides high availability for WebLogic Servers in a cluster.

Supported software

The Symantec ApplicationHA agent for WebLogic Server supports the following software versions in a VCS environment.

Operating Systems	Red Hat Enterprise Linux 5.0 on x86_84 architecture
WebLogic Server	9 and 10, including minor releases.

About WebLogic Server

WebLogic Servers fall into two categories: Administrative and Managed. The Administrative Server provides a central point from which you can manage the domain, and it provides access to WebLogic server administration tools [WLS05: *Introduction to Oracle WebLogic server and Oracle WebLogic Express*, July 2005]. All other servers are considered as Managed Servers.

A Node Manager is a WebLogic server utility that enables you to start, shut down, and restart Administration Server and Managed Server instances from a remote location.

The Symantec ApplicationHA agent for WebLogic Server supports both Administrative and Managed Servers, and Node Manager based configurations. The agent recognizes the startup server dependency that exists between Managed and Administrative Servers and provides the user with the choice of enforcing or not enforcing this startup restriction.

WebLogic 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 :

- Performs a preliminary check to ensure that the WebLogic Server component is not already running.
- Checks the value of the ServerRole attribute set for the resource. If the value of the attribute is Managed, the online function may delay the Managed server startup process until the Administrative server is initialized. For details, refer to description of attributes AdminServerMaxWait and RequireAdminServer.
- Starts the WebLogic Server component using the following mechanism.

Node Manager	Uses the wlst command `startNodeManager`.
Administrative server (NM)	Uses the wlst commands `nmConnect` and `nmStart`.

Managed server (NM)	Uses the wlst commands `nmConnect` and `nmStart`.
Administrative server (NNM)	Uses the script configured in ServerStartProgram attribute.
Managed server (NNM)	Uses the script configured in ServerStartProgram attribute.

- Ensures that the component is up and running successfully. The agent function uses the wait period that the OnlineTimeout attribute specifies, to enable the WebLogic Server component to initialize fully before allowing the monitor function to probe the newly running server instance.

Offline

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

- Performs a preliminary check to ensure that the WebLogic Server component is not already offline.
- For different resource configurations, stops the WebLogic Server component gracefully using the mechanism shown as follows.

Node Manager	Terminates the Node Manager process.
Administrative server (NM)	Uses the wlst commands `connect` and `shutdown`.
Managed server (NM)	Uses the wlst commands `connect` and `shutdown`.
Administrative server (NNM)	Uses the script configured in ServerStopProgram attribute.
Managed server (NNM)	Uses the script configured in ServerStopProgram attribute.

- Ensures that the resource is given enough time to go offline successfully. The agent function uses a wait period that the OfflineTimeout attribute specifies, to allow the WebLogic Server component to complete the offline sequence before allowing further probing of the resource.

Monitor

The monitor function performs the following tasks:

- Conducts a first level check on the WebLogic Server component to ensure that the WebLogic Server component's process is running. The agent identifies the

process for the WebLogic Server component by applying the pattern matching on command lines of processes running in the system.

- Depending on the configuration, the monitor function can conduct a second level check on the WebLogic Server component.
  The second level check uses the `wlst.sh` scripting utility to attempt to connect to the WebLogic Server component.
  For different resource configurations, the wlst commands used to connect to the WebLogic Server component are listed as follows.

Node Manager	Uses the wlst command `nmConnect`.
Administrative server (NM)	Uses the wlst command `connect`.
Managed server (NM)	Uses the wlst command `connect`.
Administrative server (NNM)	Uses the wlst command `connect`.
Managed server (NNM)	Uses the wlst command `connect`.

- Depending upon the value of the MonitorProgram attribute, the monitor function can perform a customized check using a user-supplied monitoring utility.

Note: To configure second level monitoring, use CLI.

ApplicationHA wizards configure WebLogic 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 ApplicationHA agent for WebLogic Server" on page 53.

Clean

The clean function performs the following tasks:

- Attempts to gracefully shut down the WebLogic Server component.
- For Administrative and Managed server in Node Manager based configurations, the clean function attempts the wlst `nmKill` command.
- Identifies the process for the WebLogic Server component and kills it.

The default value of the CleanTimeout attribute is 60 seconds. As the clean function may execute two wlst.sh operations, 60 seconds may be insufficient. You can set this attribute to 120 seconds or more.

Chapter 2

Setting up WebLogic Server for ApplicationHA

This chapter includes the following topics:

Executing a customized monitoring program

You can configure the monitor function to execute a custom monitor program to perform a user-defined WebLogic Server state check. Based on the UNIX user defined in the User attribute, this MonitorProgram runs in this user-defined shell.

The monitor function executes the utility specified in the MonitorProgram attribute if the following conditions are satisfied:

- The MonitorProgram attribute value is set to a valid executable program.
- The first level process check indicates that the WebLogic Server instance is online.

- The SecondLevelMonitor attribute is either set to 0 (false), or SecondLevelMonitor is set to 1 (true) and the second level check indicates that the WebLogic Server instance is online.
  This feature allows cluster administrators to define custom programs that can further determine the state of the WebLogic Server. For example, if the administrator wants to test the status of a J2EE component running inside the WebLogic Server, the administrator can execute a custom program to determine that the underlying application is working properly.

The monitor function interprets the utility exit code as follows:

110 or 0	WebLogic Server server instance is online
100 or 1	WebLogic Server server instance is offline
99	WebLogic Server server instance is unknown
Any other value	WebLogic Server server instance is unknown

To ensure that the custom monitor program is always available to the agent application, Symantec recommends storing the file in the directory that the BEA_HOME attribute specifies on the shared storage device.

Uniquely identifying WebLogic Server instances

You can virtualize a WebLogic Server instance using a cluster. It is important that the agent for WebLogic Server can uniquely identify an instance on a node that is hosting more than one simultaneous WebLogic Servers.

Differentiating WebLogic Server instances is especially important when the agent for WebLogic Server must kill the processes of a non-responsive or failed instance. Failure to define unique names for each WebLogic Server can result in a clean operation that erroneously kills processes for more than one WebLogic Server instance.

Define a unique name for each WebLogic Server as follows:

- To uniquely identify an Administrative Server instance, the combination of ServerName and DomainName must be unique for the Administrative Server instance.
- To uniquely identify a Managed Server instance, do the following:
  - The combination of ServerName and DomainName must be unique for the Managed Server instance.

- The value of the AdminUrl attribute must match the value of management server that appears in the long listing of processes for the Managed Server instance.

- To uniquely identify a Node Manager instance, the value of the nmListenAddressPort attribute must match the values of ListenAddress and ListenPort that appear in the long listing of processes for the Node Manager instance.

Attributes used in different resource configurations

For each resource configuration, some attributes may be used by the agent and others may not be used. Use the following tables to figure out which attributes must be configured for your resource depending on the required configuration for your resource.

In these tables, the following conventions hold true:

- SLM stands for SecondLevelMonitor attribute.
- "Yes" implies that attribute is mandatory for the given configuration.
- "Opt" implies that configuring the attribute is optional for the given configuration.
- "-" implies that the attribute is not used by the agent for the given configuration.

Table 2-1 shows the attributes used by Node Manager based configurations.

Table 2-1 Attributes used by Node Manager based configurations

Resource Configuration/Attributes	Node Manager (SLM=0)	Node Manager (SLM>0)	Administrative Server (NM)	Managed Server (NM)
ResLogLevel	Yes	Yes	Yes	Yes
AdminURL	-	-	-	Yes
BEA_HOME	Yes	Yes	Yes	Yes
WL_HOME	Yes	Yes	Yes	Yes
DomainName	-	Yes	Yes	Yes
DomainDir	-	Yes	Yes	Yes
ListenAddressPort	-	-	Yes	Yes

Table 2-1 Attributes used by Node Manager based configurations *(continued)*

Resource Configuration/Attributes	Node Manager (SLM=0)	Node Manager (SLM>0)	Administrative Server (NM)	Managed Server (NM)
MonitorProgram	Opt	Opt	Opt	Opt
nmListenAddressPort	Yes	Yes	Yes	Yes
nmType	Yes	Yes	Yes	Yes
ServerName	-	-	Yes	Yes
ServerRole	Yes	Yes	Yes	Yes
WLSUser	Yes	Yes	Yes	Yes
WLSPassword	Yes	Yes	Yes	Yes
RequireAdminServer	-	-	-	Yes
AdminServerMaxWait	-	-	-	Yes
SecondLevelMonitor	0	> 0	Yes	Yes
ServerStartProgram	-	-	-	-
ServerStopProgram	-	-	-	-
User	Yes	Yes	Yes	Yes

Table 2-2 shows the attributes used by non-Node Manager based configurations.

Table 2-2 Attributes used by non-Node Manager based configurations

Resource Configuration/Attributes	Managed Server (NNM) (SLM=0)	Managed Server (NNM) (SLM>0)	Administrative Server (NNM) (SLM=0)	Administrative Server (NNM) (SLM>0)
ResLogLevel	Yes	Yes	Yes	Yes
AdminURL	Yes	Yes	-	-
BEA_HOME	Yes	Yes	Yes	Yes
WL_HOME	Yes	Yes	Yes	Yes
DomainName	Yes	Yes	Yes	Yes

Table 2-2 Attributes used by non-Node Manager based configurations *(continued)*

Resource Configuration/Attributes	Managed Server (NNM) (SLM=0)	Managed Server (NNM) (SLM>0)	Administrative Server (NNM) (SLM=0)	Administrative Server (NNM) (SLM>0)
DomainDir	Yes	Yes	Yes	Yes
ListenAddressPort	Yes	Yes	Yes	Yes
MonitorProgram	Opt	Opt	Opt	Opt
nmListenAddressPort	-	-	-	-
nmType	-	-	-	-
ServerName	Yes	Yes	Yes	Yes
ServerRole	Yes	Yes	Yes	Yes
WLSUser	-	Yes	-	Yes
WLSPassword	-	Yes	-	Yes
RequireAdminServer	Yes	Yes	-	-
AdminServerMaxWait	Yes	Yes	-	-
SecondLevelMonitor	0	> 0	0	> 0
ServerStartProgram	Yes	Yes	Yes	Yes
ServerStopProgram	Yes	Yes	Yes	Yes
User	Yes	Yes	Yes	Yes

Using WebLogic provided scripts

WebLogic built-in scripts can be used in non-Node Manager based configurations as values of ServerStartProgram and ServerStopProgram attributes. When you create a domain using the config.sh utility, WebLogic generates some scripts.

You can use the following scripts to start or stop WebLogic Server instances present in the WebLogic domain.

- To start an Administrative Server instance, use the following command:

```
# DomainDir/bin/startWebLogic.sh
```

Where *DomainDir* is the name of the directory where the domain binaries and configuration files are stored.

- To stop an Administrative Server instance, use the following command:

  ```
  # DomainDir/bin/stopWebLogic.sh
  ```

- To start a Managed server instance, use the following command:

  ```
  # DomainDir/bin/startManagedWebLogic.sh
  ```

- To stop a Managed server instance, use the following command:

  ```
  # DomainDir/bin/stopManagedWebLogic.sh
  ```

Note: A valid user name and password are required for starting and shutting down WebLogic Server when it runs in production mode. The agent requires startup and shutdown scripts to execute non-interactively. Ensure that the username and password are defined in ${DOMAIN_HOME}/bin/startManagedWebLogic.sh and ${DOMAIN_HOME}/bin/stopWebLogic.sh if it is not passed as command line arguments.

Avoiding storing unencrypted credentials in startup/shutdown scripts

Whenever you configure a WebLogic server under ApplicationHA that uses WebLogic provided scripts to start and stop the WebLogic server it is recommended to have the boot identity files to avoid storing unencrypted credentials in startup/shutdown scripts. The boot identity file boot.properties should be created for the WebLogic server and placed in the security directory of the server.

For more details, refer to

http://download.oracle.com/docs/cd/E13222_01/wls/docs90/server_start/overview.html#1068976

Note: If you do not have the boot.properties file, and have not provided the username/password to start/stop scripts, the start and stop scripts will prompt you for a username and password. If ApplicationHA invokes the start or stop operation, this prompt causes the operation to fail.

Delaying managed server startup process

WebLogic Managed Servers initiate a connection to the Administrative Server while trying to download configuration information.

If ApplicationHA starts up all the WebLogic Servers in virtual machine at the same time, delaying the startup process of Managed Servers until the Administrative Server is fully initialized is advantageous. You can set the AdminServerMaxWait attribute to orchestrate such a delay.

The online function uses the AdminServerMaxWait attribute to control a repeating cycle of probe, wait, probe, and wait until the presence of the Administrative Server is detected successfully. After the server is fully initialized, the online function proceeds with the Managed Server startup.

If the Administrative Server is not available before the wait time expires, the online function generates a cluster log warning message and proceeds with instance startup.

You can control the Managed Server delaying process in the following ways:

- If the RequireAdminServer attribute is set to 1 (true), the online function does not proceed until the Administrative Server is available and ready to accept connections. If the time spent waiting on the availability of the Administrative Server exceeds the value of OnlineTimeout, the online function generates an error message indicating the source of the problem and terminates.
- If the RequireAdminServer attribute is set to 0 (false) and the AdminServerMaxWait attribute is set to a number greater than zero, the online procedure waits up to AdminServerMaxWait seconds for the Administrative Server to transition to a running state before proceeding with the online procedure. If the time spent waiting on the availability of the Administrative Server exceeds the value of AdminServerMaxWait, the online function proceeds with the remaining online steps and does not wait for the availability of an Administrative Server.

The online function interprets the AdminServerMaxWait attribute value as follows:

Value	Interpretation
0 - 5	Wait the specified number of seconds, then immediately start the online procedures. Do not check to see if the Admin Server is ready.
6 - ($NSR-3)	Wait the specified number of seconds, then check to see if the Admin Server is ready. $NSR represents the number of seconds remaining before the OnlineTimeout would be reached.

> ($NSR-3)	A value greater than the $NSR (minus 3) causes the agent for WebLogic to wait up to three seconds before the OnlineTimeout is about to expire, and to insert an info-level message into the cluster log file.

Configuring multiple Administrative Servers having the same name but different domains for Non Node Manager based configurations

When you configure ApplicationHA with multiple administrative WebLogic Server servers having the same server name but different domain names, the agent needs to verify that the process list output of the WebLogic instance contains the environment variable "DOMAIN_HOME" with the value specified for DomainDir during configuration. If this environment variable is present in the process output, the resource will identify all the administrative servers for different domains separately.

But if this environment variable is not present in the process output of the instance, the WebLogic supplied start script, $DomainDir/bin/startWebLogic.sh will need to be modified.

Add the "-Dwl.domain=<*domainName*>" in the java command which starts the WebLogic Server, where *domainName* needs to be replaced with the name of the particular domain, for all the domains with same administrative server name.

To modify the script, do the following:

- Create a copy of the DomainDir/bin/startWebLogic.sh file
- Rename the copy as DomainDir/bin/startWebLogic_old.sh
- In the startWebLogic.sh file, replace the following lines:

```
 echo "${JAVA_HOME}/bin/java ${JAVA_VM} ${MEM_ARGS}
 ${JAVA_OPTIONS} -Dweblogic.Name=${SERVER_NAME}
-Djava.security.policy=${WL_HOME}/server/lib/weblogic.policy
${PROXY_SETTINGS} ${SERVER_CLASS}"
${JAVA_HOME}/bin/java ${JAVA_VM} ${MEM_ARGS} ${JAVA_OPTIONS}
-Dweblogic.Name=${SERVER_NAME} -Djava.security.policy=${WL_HOME}
/server/lib/weblogic.policy ${PROXY_SETTINGS} ${SERVER_CLASS}
```

  with the following lines:

```
echo "${JAVA_HOME}/bin/java ${JAVA_VM} ${MEM_ARGS}
${JAVA_OPTIONS} -Dweblogic.Name=${SERVER_NAME}
-Djava.security.policy=${WL_HOME}/server/lib/weblogic.policy
```

```
-Dwl.domain=<domainName> ${PROXY_SETTINGS} ${SERVER_CLASS}"
${JAVA_HOME}/bin/java ${JAVA_VM} ${MEM_ARGS} ${JAVA_OPTIONS}
-Dweblogic.Name=${SERVER_NAME} -Djava.security.policy=${WL_HOME}
/server/lib/weblogic.policy -Dwl.domain=<domainName>
${PROXY_SETTINGS} ${SERVER_CLASS}
```

Chapter 3

Installing and removing the WebLogic Server agent

This chapter includes the following topics:

Installing and removing the WebLogic Server

The WebLogic Server is installed and removed as part of the ApplicationHA installation and uninstallation.

For more information, see the *Symantec ApplicationHA Installation Guide*.

Chapter 4

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 VMware 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 the **Configure Application Monitoring** link on the ApplicationHA tab in 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, from the command line.
  Use the VCS commands to configure application service groups. You must ensure that the resource names are unique. Refer to the *Veritas Cluster Server Administrator's Guide* for information on commands. You can find it here:
  https://vos.symantec.com/documents/doc_details/sfha/5.1/Linux/ProductGuides/
- After configuring WebLogic Server for monitoring, if you create another WebLogic server instance, this new instance is not monitored as part of the existing configuration.
  In such a case, you must first unconfigure the existing monitoring configuration and then reconfigure the application using the wizard. You can then select all the instances for monitoring.

Before configuring application monitoring for WebLogic Server

Ensure that you complete the following tasks before configuring application monitoring for WebLogic 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 4.1
- 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 WebLogic Server and 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 Configuration Guide* for a list of ports and services used.

Configuring application monitoring for WebLogic Server

Perform the following steps to configure monitoring for WebLogic Server on a virtual machine hosted on a VMware vCenter Server managed ESX 4.1 Server.

You can configure WebLogic Server instances for monitoring with node-manager-based specifications or non-node manager-based specifications.

By using Node Manager specifications

To configure application monitoring for WebLogic Server using Node Manager specifications.

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 WebLogic 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 **WebLogic Server** and then click **Next**.

   The WebLogic Server Domain Directory Specification screen appears.

7. On the WebLogic Server Domain Directory Specification screen, enter the path where the WebLogic server domains are installed on the virtual machine and then click **Next**.

   The WebLogic Server Instance Selection screen for node manager instances appears.

8 Select node manager/s and enter the appropriate values in the following fields.

WLS User Name	The user name of the user that is connecting the wlst.sh utility to the server running the WebLogic Server instance.
WLS Password	The password of the user that is connecting the wlst.sh utility to the server running the WebLogic Server instance.
User	The UNIX user name used to start and stop the WebLogic Server instance.
Node Manager Listen Address	The Listen Address of the WebLogic Node Manager.
Node Manager Listen Port	The port of the WebLogic Node Manager.
Node Manager Communication Type	The WebLogic Node Manager type. This type is used while connecting to the Node Manager through the wlst.sh script.

9 Click **Next**

The WebLogic Server Instance Selection screen for administrative servers and managed servers appears along with node manager details.

10 Select the administrative server/s and managed server/s, specify the WebLogic Server (WLS) user credentials, and then click **Next**.

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

11 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.

---

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

This completes the application monitoring configuration.

13 To view the application status, click **Refresh**, and then open the ApplicationHA view in vSphere Client. The view displays the application as configured and running on the virtual machine.

Without using Node Manager specifications

To configure application monitoring for WebLogic Server without using Node Manager specifications.

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 WebLogic 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 **WebLogic Server** and then click **Next**.

   The WebLogic Server Domain Directory Specification screen appears.

7. On the WebLogic Server Domain Directory Specification screen, enter the path where the WebLogic server domains are installed on the virtual machine and then click **Next**.

   The WebLogic Server Instance Selection screen for node manager instances appears.

   Note: If Node Managers are configured on the virtual machine, the screen will list them. Without selecting any node manager/s, perform this step.

8. Click **Next**.

   The WebLogic Server Instance Selection screen for administrative servers and managed servers appears.

9 Select the administrative server/s and managed server/s, enter the appropriate information in the following fields, and then click **Next**.

WLS User Name	The user name of the user that is connecting the wlst.sh utility to the server running the WebLogic Server instance.
WLS Password	The password of the user that is connecting the wlst.sh utility to the server running the WebLogic Server instance.
User	The UNIX user name used to start and stop the WebLogic Server instance.
Start Program	The complete command line of the script used to start WebLogic Server.
Stop Program	The complete command line of the script used to stop WebLogic Server.

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

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

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

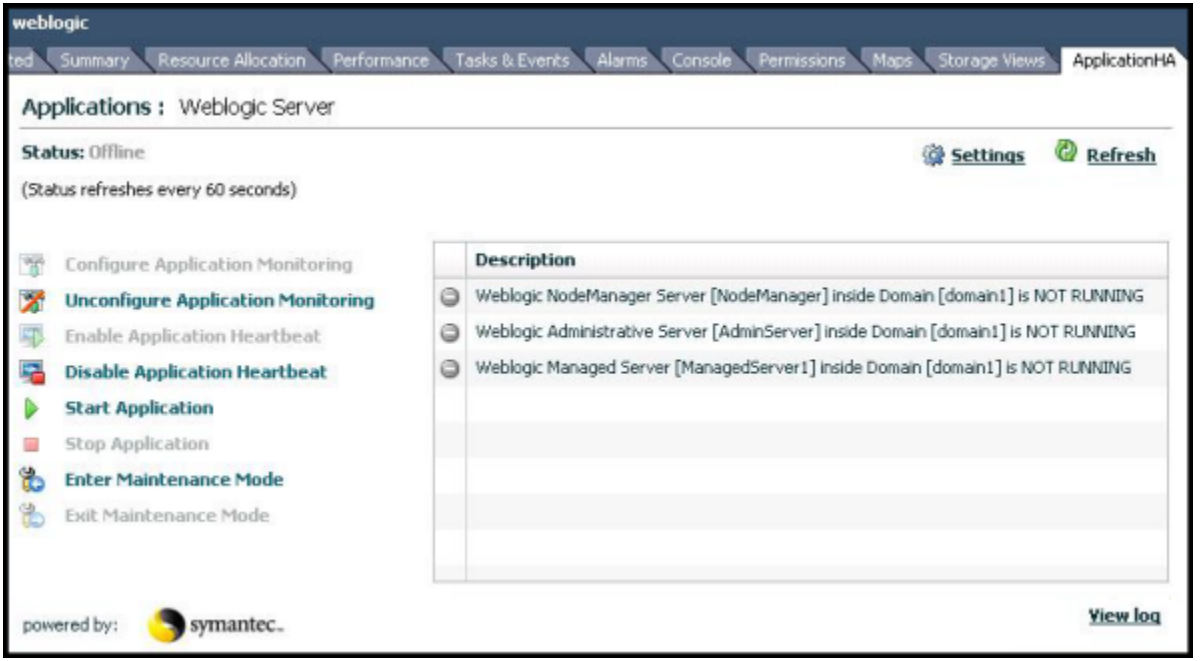
You then have to run the wizard again to configure the application monitoring.

---

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

This completes the application monitoring configuration.

12 To view the application status, click **Refresh**, and then open the ApplicationHA view in vSphere Client. The view displays the application as configured and running on the virtual machine.



Chapter 5

Troubleshooting the agent for WebLogic Server

This chapter includes the following topics:

Starting the WebLogic Server instance outside the Symantec ApplicationHA environment

If you face problems while working with a WebLogic Server, you must click **Enter Maintenance Mode**.

You can then restart the WebLogic Server instance outside the ApplicationHA environment.

Note: Use the same parameters that you specified while configuring the WebLogic Server under ApplicationHA.

A sample procedure to start a WebLogic Server instance outside the ApplicationHA environment, is illustrated as follows.

To restart a Node Manager outside the ApplicationHA environment

1. Log in as superuser onto the host on which the WebLogic Node Manager application is to run.
2. Use the values defined in the agent attributes to initiate the Node Manager start program.

   For example, assume that the following values are assigned:

Attribute	Value
User	weblogic
BEA_HOME	/bea/wls90/admin
nmListenAddressPort	wls90admsol:5556
nmType	ssl
ServerRole	NodeManager
WL_HOME	/bea/wls90/admin/weblogic90

3. Log in to the Node Manager using the user name specified in the User attribute:

   ```
   # su - weblogic
   ```

4. Go to the directory specified in the BEA_HOME attribute:

   ```
   # cd /bea/wls90/admin
   ```

5. Start the WebLogic Server Scripting Tool:

   ```
   # /bea/wls90/admin/weblogic90/common/bin/wlst.sh
   ```

6 Start the Node Manager:

```
# startNodeManager(verbose='true',NodeManagerHome='/bea/wls90/
 admin/weblogic90/common/nodemanager',
 ListenPort='5556',ListenAddress='wls90admsol')
```

If the Node Manager starts successfully, following message is displayed:

```
Successfully launched the Node Manager.
```

7 Enter this command:

```
# exit()
```

If the Node Manager works properly outside the ApplicationHA framework, you can then attempt to implement the Node Manager within the ApplicationHA environment .

To restart a Managed or Administrative Server outside the ApplicationHA environment

1 Log in as superuser in to the host on which the WebLogic Server application is to run.

2 Use the values defined in the agent attributes to initiate the WebLogic Server start program.

For example, for an Administrative Server, assume that the following values are assigned:

Attribute	Value
ServerName	AdminServer
ServerRole	Administrative
BEA_HOME	/bea/wls90/admin
DomainName	WLS90Domain
nmListenAddressPort	wls90admsol:5556
WL_HOME	/bea/wls90/admin/weblogic90
DomainDir	/bea/wls90/admin/user_projects/domains/WLS90Domain
nmType	ssl
User	weblogic

3 Log in to the Administrative Server using the user name specified in the User attribute:

```
# su - weblogic
```

4 Go to the directory specified in the BEA_HOME attribute:

```
# cd /bea/wls90/admin
```

5 Start the WebLogic Server Scripting Tool:

```
# /bea/wls90/admin/weblogic90/common/bin/wlst.sh
```

6 Connect to the Node Manager:

```
# nmConnect('weblogic', 'asdf1234', 'wls90adminsol','5556',
'WLS90Domain', '/bea/wls90/admin/user_projects/domains/
WLS90Domain','ssl')
```

7 Start the Administrative Server:

```
# nmStart("AdminServer")
```

If the server starts successfully, the following message is displayed:

```
Starting Server AdminServer
Server AdminServer started successfully
```

If the WebLogic Server works properly outside the ApplicationHA framework, you can then attempt to implement the server within the ApplicationHA framework.

Reviewing error log files

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

Using WebLogic Server log files

If the WebLogic Server is facing problems, access the log files of the WebLogic Server to further investigate the problem. The log files are located as follows:

- For Node Managers:

  WL_HOME/common/nodemanager/nodemanager.log

- For Administrative Servers:

  DomainDir/servers/*ServerName*/*ServerName*.log
  DomainDir/servers/*ServerName*/*ServerName*.out

- For Managed Servers:

  DomainDir/servers/*ServerName*/*ServerName*.log
  DomainDir/servers/*ServerName*/*ServerName*.out
  DomainDir/servers/*ServerName*/access.log

Reviewing ApplicationHA log files

In case of problems while using the agent for WebLogic Server, you can access the ApplicationHA log files at the following location:

/var/VRTSvcs/log

Using trace level logging

The ResLogLevel attribute controls the level of logging that is written in a ApplicationHA log file for each WebLogic 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 resource.

To localize ResLogLevel attribute for a resource

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

   # /opt/VRTSvcs/bin/hares -local *Resource_Name* ResLogLevel

3. Set the ResLogLevel attribute to TRACE for the identified resource:

   # /opt/VRTSvcs/bin/hares -modify *Resource_Name* ResLogLevel TRACE -sys *S*

4. Note the time before you begin to operate the identified resource.
5. Test the identified resource. The function reproduces the problem that you are attempting to diagnose.
6. Note the time when the problem is reproduced.

7 Set the ResLogLevel attribute back to INFO for the identified resource:

```
# /opt/VRTSvcs/bin/hares -modify Resource_Name ResLogLevel INFO -sys SysA
```

8 Review the contents of the log file. Use the time noted in Step 4 and Step 6 to diagnose the problem.

Using agent for WebLogic Server log files

In case of problems while using the agent for WebLogic Server, you can access the agent log files for more information. The agent saves output of every operation 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.out
```

For example:

```
/tmp/.VRTSWebLogic9/WLS90Mng01_nodemanager_online.out
/tmp/.VRTSWebLogic9/WLS90Mng01_nodemanager_offline.out
/tmp/.VRTSWebLogic9/WLS90Mng01_nodemanager_clean.out
/tmp/.VRTSWebLogic9/WLS90Mng01_nodemanager_monitor.out
```

If a resource, WLS90Mng01_nodemanager is unable to bring a WebLogic Node Manager online, you can access the /tmp/.VRTSWebLogic9/WLS90Mng01_nodemanager_online.out for more information so that you can diagnose the problem.

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

Problems starting a Managed Server through the administrative console

You may encounter problems while starting a Managed Server through the Administrative console. When you start a Managed server through the console, the Administrative Server sends a request to the Node Manager to start the Managed Server. The Administrative Server sends this request using SSL communication.

If the Node Manager is running on a virtual host, this communication may fail. This failure may occur because the Node Manager uses default SSL certificates

that contain the real host name of the physical node on which the Node Manager is running. The URL used for connecting to the Node Manager contains the virtual host name of the Node Manager, which is different from the physical host name of the node. The Administrative Server rejects the communication because of this mismatch.

To overcome this mismatch, you can perform one of the following procedures:

- Generate new SSL certificates
  You can generate new SSL certificates that contain the virtual host name of the Node Manager. Then, configure the Node Manager to use the new SSL certificates.

  For more details about creating SSL certificates, refer to the following links:

  - http://download.oracle.com/docs/cd/E13222_01/wls/docs90/secmanage/ssl.html
  - http://download.oracle.com/docs/cd/E13222_01/wls/docs90/server_start/nodemgr.html
  - http://download.oracle.com/docs/cd/E13222_01/wls/docs90/secmanage/identity_trust.html

  Oracle recommends generating new SSL certificates using reliable certification authorities as best security practice. Otherwise, you can generate certificates and keystores which use virtual hostname, using the tools, CertGen and ImportPrivateKey that WebLogic provides.

- Disable the host name verification function
  You can disable the host name verification function in the Administrative Server properties. For details about disabling the function, refer to the following link:
  http://download.oracle.com/docs/cd/E13222_01/wls/docs90/ConsoleHelp/taskhelp/security/DisableHostNameVerification.html

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 explain the attributes associated with the agent. The Required attributes explain the attributes that must be configured for the agent to function properly.

Resource type definition for WebLogic Server agent

Examples of agent type definition files are:

```
type WebLogic (
static keylist LogDbg = { DBG_21 }
static i18nstr ArgList[] = { ResLogLevel, State, IState,
AdminURL, BEA_HOME, WL_HOME, DomainName, DomainDir,
ListenAddressPort, MonitorProgram, nmListenAddressPort, nmType,
ServerName, ServerRole, WLSUser, WLSPassword, RequireAdminServer,
AdminServerMaxWait, SecondLevelMonitor, ServerStartProgram,
ServerStopProgram }
str ResLogLevel = INFO
str AdminURL
str BEA_HOME
str WL_HOME
str DomainName
str DomainDir
```

```
str ListenAddressPort
str MonitorProgram
str nmListenAddressPort
str nmType = ssl
str ServerName
str ServerRole
str WLSUser
str WLSPassword
boolean RequireAdminServer = 0
int AdminServerMaxWait = 60
int SecondLevelMonitor
str ServerStartProgram
str ServerStopProgram
)
```

Attribute definition for WebLogic Server agent

Refer to the following required and optional attributes while configuring the agent for WebLogic Server.

Table A-1 lists the required attributes for the agent for WebLogic Server.

Table A-1 Required attributes

Required attribute	Description
BEA_HOME	The absolute path to BEA home directory of WebLogic Server installation. BEA_HOME is used to uniquely identify the ServerRole processes. Type and dimension: string-scalar Default: "" Example: /bea/wls90/admin

Table A-1 Required attributes *(continued)*

Required attribute	Description
DomainDir	The domain directory of the WebLogic Server domain to which the instance belongs. The agent for WebLogic Server uses this attribute to connect to the Node Manager using the wlst.sh utility. Specify this attribute for Administrative and Managed Servers. If the SecondLevelMonitor attribute is specified, specify this attribute for the Node Manager also. Type and dimension: string-scalar Default: "" Example: /bea/wls90/admin/user_projects/domains/WLS90Domain
DomainName	The name of the WebLogic Server domain to which the instance belongs. The WebLogic Server uses this attribute to connect to the Node Manager using the wlst.sh utility. Specify this attribute for Administrative and Managed Servers. If the SecondLevelMonitor attribute is specified, specify this attribute for the Node Manager also. See "Uniquely identifying WebLogic Server instances" on page 16. Type and dimension: string-scalar Default: "" Example: WLS90Domain
ListenAddressPort	The Listen Address and port of the WebLogic instance. The format is ListenAddress:port. Ensure that the ListenAddress string resolves to the proper IP Address, using the network name service that you used on the host. The WebLogic Server connects to the ListenAddress on the specified port through the wlst.sh API. Specify this attribute for Administrative and Managed Servers only. Type and dimension: string-scalar Default: "" Example: wls90adminsol.veritas.com:7001 or wls90adminsol.veritas.com:5556

Table A-1 Required attributes *(continued)*

Required attribute	Description
nmListenAddressPort	The Listen Address and port of the WebLogic Node Manager. The format is ListenAddress:port. The value of this attribute must match the values of ListenAddress and ListenPort that appear in the long listing of processes for a Node Manager instance. The ListenAddress string must resolve to a proper IP Address, using the network name service that you used on the host. The agent for WebLogic Server uses the ListenAddress on the specified port to connect through the wlst.sh API. Type and dimension: string-scalar Default: "" Example: wlsadmin:5556
nmType	The WebLogic Node Manager type. This type is used while connecting to the Node Manager through the wlst.sh script. Valid values include: ■ plain: plain socket Java-based implementation ■ rsh: RSH implementation ■ ssh: script-based SSH implementation ■ ssl: Java-based SSL implementation Type and dimension: string-scalar Default: ssl Example: ssh
ResLogLevel	The logging detail performed by the agent for WebLogic Server for the resource. Valid values are: ERROR: Only logs error messages. WARN: Logs above plus warning messages. INFO: Logs above plus informational messages. TRACE: Logs above plus trace messages. TRACE is very verbose and should only be used during initial configuration or for troubleshooting and diagnostic operations. Type and dimension: string-scalar Default: INFO Example: TRACE

Table A-1 Required attributes *(continued)*

Required attribute	Description
ServerName	The name of the WebLogic Server. You must specify this attribute for Administrative and Managed Servers only. See "Uniquely identifying WebLogic Server instances" on page 16. Type and dimension: string-scalar Default: "" Example: AdminServer
WL_HOME	The absolute path to the product installation directory of the WebLogic Server. The agent for WebLogic Server uses this attribute to locate the wlst.sh utility and the Node Manager home directory. Type and dimension: string-scalar Default: "" Example: /bea/wls90/admin/weblogic90
WLSUser	The user name of the user that is connecting the wlst.sh utility to the server running the WebLogic Server instance, along with WLSPassword. Type and dimension: string-scalar Default: ""

Table A-1 Required attributes *(continued)*

Required attribute	Description
ServerRole	Type of WebLogic Server. Valid values are: ■ NodeManager: Online operation executes wlst.sh script with startNodeManager() API. Example: startNodeManager(verbose='true', NodeManagerHome='/bea/wls90/admin/weblogic90/common/nodemanager',ListenPort='5556', ListenAddress='wls90adminsol') ■ Administrative: Online operation executes wlst.sh script with nmConnect() and nmStart() API. Example: nmStart (‘AdminServer1’) ■ Managed: Online operation executes wlst.sh script with nmConnect() and nmStart() API. Example: nmStart (‘ManagedServer1’) Type and dimension: string-scalar Default: "" Example: Administrative
User	The UNIX user name used to start and stop the WebLogic Server instance. If MonitorProgram is specified, the agent for WebLogic Server uses this user's credentials to run the defined program. You must synchronize the user name across the systems within the cluster. This user name must resolve to the same UID and have the same default shell on each system in the cluster. The agent operations use the getpwname(3C) function system call to obtain UNIX user attributes. Hence you can define the user name locally or in a common repository such as NIS, NIS+, or LDAP. Type and dimension: string-scalar Default: "" Example: wlsadmin

Table A-1 Required attributes *(continued)*

Required attribute	Description
WLSPassword	The password of user connecting WLST to ServerRole Application Server, along with WLSUser. Encrypt the value of this attribute using the /opt/bin/vcsencrypt utility that ApplicationHA provides. While encrypting the password, use the -agent option. Type and dimension: string-scalar Default: "" Example: weblogic
ServerStartProgram	The complete command line of the script used to start WebLogic Server. If command line arguments are passed to ServerStartProgram, the agent uses the command and arguments as it is. Example: ServerStartProgram = "/wls/my_domain/startManagedWebLogic.sh Managed1" If no arguments are passed (for example, ServerStartProgram = "/wls/my_domain/startManagedWebLogic.sh"), the agent forms the command line as follows: ■ For Managed Server: `$ServerStartProgram $ServerName $AdminURL` ■ For Administrative Server: `$ServerStartProgram` Type and dimension: string-scalar Default: "" Example: /bea/user_projects/domains/WLS90Domain/bin/startManagedWebLogic.sh

Table A-1 Required attributes *(continued)*

Required attribute	Description
ServerStopProgram	The complete command line of the script used to stop WebLogic Server. If command line arguments are passed to ServerStopProgram, the agent uses the command and arguments as it is. Example: ServerStopProgram = "/wls/my_domain/stopManagedWebLogic.sh Managed1 t3://adminurl:7001 weblogic passwd" If no arguments are passed (for example, ServerStopProgram = "/wls/my_domain/stopManagedWebLogic.sh", the agent forms the command line as follows: `$ServerStopProgram $ServerName $AdminURL $WLSUser $WLSPassword` Type and dimension: string-scalar Default: "" Example: /bea/user_projects/domains/WLS90Domain/bin/stopManagedWebLogic.sh

Table A-2 lists the optional attributes.

Table A-2 Optional attributes

Optional attribute	Description
AdminUrl	The URL of the Managed Server's Administrative Server. Set this attribute only for resources whose ServerRole attribute is Managed. Ensure that the value of this attribute is the same as management server that appears in the long listing of processes for the Managed Server. If the RequireAdminServer attribute is set to 1, AdminUrl is used to connect to the Administrative Server for the domain to determine if the server is fully online. Managed Servers also use this URL to connect to the Administrative Server and download its web applications and services (JMS, JDBC Connection Pool, etc.) configuration. Type and dimension: string-scalar Default: "" Example: http://wlsadmin:7001

Table A-2 Optional attributes *(continued)*

Optional attribute	Description
AdminServerMaxWait	The maximum number of seconds that a Managed Server waits for an Administrative Server to respond to a test probe. See "Delaying managed server startup process" on page 21. Type and dimension: integer-scalar Default: 60 Example: 90
MonitorProgram	The full pathname and command-line arguments for an externally provided monitor program. See "Executing a customized monitoring program" on page 15. Type and dimension: string-scalar Default: "" Example 1: /bea/wls90/admin/mymonitor.sh Example 2: /usr/local/bin/MyMonitor.sh myWLS.foo.com 8080
RequireAdminServer	The flag that is used to control the startup behavior of a WebLogic Server instance. When the RequireAdminServer attribute is set to 1 (true), the Managed Server resource is not allowed to complete an initiated online operation until the Administrative Server is ready to accept connections. If the RequireAdminServer attribute is set to 0 and the AdminServerMaxWait is set to a value > 5, the online operation first probes the Administrative Server instance to see if it is ready to accept connections. If the server is not ready, the operation waits for 5 seconds and then probes the server again to determine its state. This cycle of probe and wait repeats until either the Administrative Server is ready or the AdminServerMaxWait time expires. Specify this attribute for Managed Server only. Type and dimension: boolean-scalar Default: 0 (false) Example: 1 (true)

Table A-2 Optional attributes *(continued)*

Optional attribute	Description
SecondLevelMonitor	Used to enable second-level monitoring. Second-level monitoring is a deeper, more thorough state check of the configured ServerRole. The numeric value specifies how often the monitoring routines must run. ■ 0 means never run the second-level monitoring routines ■ 1 means run routines every monitor interval ■ 2 means run routines every second monitor interval, and so on. The agent for WebLogic Server uses the BEA supplied WebLogic Server scripting tool wlst.sh, to perform second-level monitoring. Depending upon the ServerRole, wlst.sh uses api commands connect(), nmConnect() and nmServerStatus() to perform monitoring routines. **Note:** Exercise caution while setting SecondLevelMonitor to large numbers. For example, if the MonitorInterval is set to 60 seconds and the SecondLevelMonitor is set to 100, then wlst.sh is executed every 100 minutes, which may not be as often as intended. For maximum flexibility, no upper limit is defined for SecondLevelMonitor. Type and dimension: integer-scalar Default: 0 Example: 1

Appendix B

Detail monitoring

This appendix includes the following topics:

Setting the PATH variable

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 WebLogic Server

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

To enable detail monitoring for WebLogic 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 Weblogic_<DomainName>_SG
   ```

3. Enable detail monitoring for WebLogic Server resources by using the following ApplicationHA commands:

   ```
   hares -modify Weblogic_<DomainName>_<ServerName>_res
   SecondLevelMonitor <frequency>
   ```

   Note: For more information on SecondLevelMonitor attribute, See "Attribute definition for WebLogic Server agent " on page 44.

4. Save the configuration and unfreeze the service group.

   ```
   haconf -dump -makero
   ```

   ```
   hagrp -unfreeze Weblogic_<DomainName>_SG
   ```

To disable detail monitoring for WebLogic 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 Weblogic_<DomainName>_SG
   ```

3. Enable detail monitoring for SAP resources by using the following ApplicationHA commands:

   ```
   hares -modify Weblogic_<DomainName>_<ServerName>_res
   SecondLevelMonitor 0
   ```

4. Save the configuration and unfreeze the service group.

   ```
   haconf -dump -makero
   ```

   ```
   hagrp -unfreeze Weblogic_<DomainName>_SG
   ```