Veritas High Availability Agent 5.0 for WebLogic Server Installation and Configuration Guide

Windows 2000, Windows 2003

11300265

symantec.

Veritas High Availability Agent 5.0 for WebLogic Server
Installation and Configuration Guide

5.0

Symantec Corporation
20330 Stevens Creek Blvd.
Cupertino, CA 95014
www.symantec.com

Third-party legal notices

Third-party software may be recommended, distributed, embedded, or bundled with this Symantec product. Such third-party software is licensed separately by its copyright holder. All third-party copyrights associated with this product are listed in the accompanying release notes.

Windows is a registered trademark of Microsoft Corporation.

Technical support

For technical assistance, visit http://support.veritas.com and select phone or email support. Use the Knowledge Base search feature to access resources such as TechNotes, product alerts, software downloads, hardware compatibility lists, and our customer email notification service.

Contents

Chapter 1

Introducing the Veritas Agent for WebLogic Server

Welcome to the Veritas high availability agent for WebLogic Server.

This guide describes the agent, agent operations, and agent attributes. The guide assumes that the reader understands the primary components and basic functionality of Veritas Cluster Server (VCS). It also assumes a basic understanding of the WebLogic Server architecture and its configuration options.

This chapter contains the following sections:

Supported software

The Veritas high availability agent for WebLogic Server is supported in the following environments:

Environment	Supported Versions
Veritas Cluster Server	4.2, 4.3
Operating Systems	Windows 2000 Server Windows 2003 Server (32-bit)
WebLogic Server	9.0, 9.1, 9.2

About the agent for WebLogic Server

The Veritas high availability agent for WebLogic Server is named WebLogic9. It consists of a resource type declaration and the agent DLL. The agent is responsible for starting, stopping, monitoring, and detecting failures of WebLogic Server (WLS) components.

About the WebLogic Server

A WebLogic Server (WLS) domain is a logical organization of WebLogic servers. The WLS domain consists of an Administrative server and one or more Managed servers.

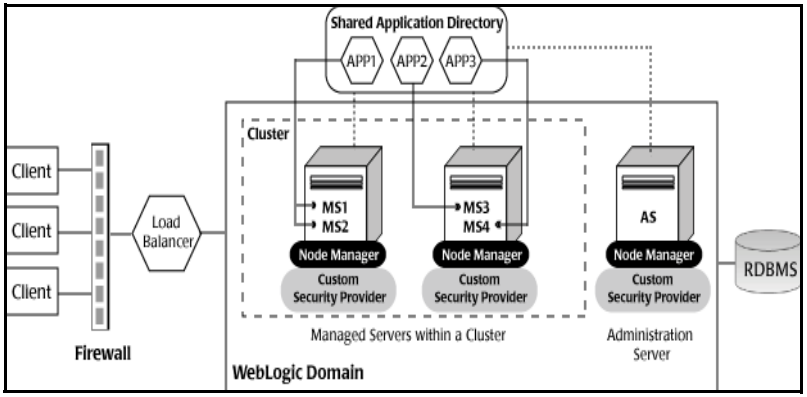
These components are described as follows:

- *Administrative server*—An Administrative server is a J2EE application server that provides centralized administration for a WLS domain.
- *Managed server*—A Managed server is a J2EE application server that hosts J2EE applications, components, and resources.

A Node Manager is a process that controls all WebLogic Server instances running on a single system or a virtual machine. The Node Manager can detect server failure and can restart the server almost instantaneously.

Figure 1-1 shows a typical WLS domain setup.

Figure 1-1 Typical WLS domain setup



The agent is WebLogic Cluster agnostic. In other words, this agent can provide high availability for stand-alone WebLogic servers and can support Managed servers that participate in a WebLogic Cluster.

Resource configurations supported by the agent

The agent supports the following kinds of resource configurations:

- Node Manager
- Administrative server: Node Manager based configuration (NM)
  In this configuration, the agent directs the Node Manager to start the Administrative server.
- Managed server: Node Manager based configuration (NM)
  In this configuration, the agent directs the Node Manager to start the Managed server.
  The advantage of Node Manager based configurations for WebLogic servers is that the Node Manager is capable of detecting server failure using internal protocol and restarting it almost instantaneously.
- Administrative server: non-Node Manager based Configuration (NNM)
  In this configuration, the agent uses custom or WebLogic provided scripts configured by the user in the ServerStartProgram and ServerStopProgram attributes to start and stop the Administrative server.
- Managed server: non-Node Manager Configuration (NNM)
  In this configuration, the agent uses custom or WebLogic provided scripts configured by the user in the ServerStartProgram and ServerStopProgram attributes to start and stop the Managed server.

The agent distinguishes between Node Manager and non-Node Manager based configurations based on whether the ServerStartProgram attribute is null or non-null. If the value is null, the agent assumes a Node Manager based configuration, otherwise it assumes a non-Node Manager based configuration.

Agent operations

The agent performs the following operations under VCS control:

- Online
- Offline
- Monitor
- Clean

The following sections elaborate the steps performed in each agent operation.

Online operation

The online operation performs the following 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 operation may delay the Managed server startup process till the Administrative server is initialized. For details, refer to description of attributes AdminServerMaxWait and RequireAdminServer.
- Starts the WebLogic Server component using the mechanism shown in Table 1-1.

Table 1-1 Mechanism to start the WebLogic Server components

Resource Configuration	Mechanism used to start the resource
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.

Table 1-1 Mechanism to start the WebLogic Server components

Resource Configuration	Mechanism used to start the resource
Managed server (NNM)	Uses the script configured in ServerStartProgram attribute.

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

Offline operation

The offline operation performs the following tasks:

- Performs a preliminary check to ensure that the WebLogic Server component is not already offline.
- Stops the WebLogic Server component gracefully using the mechanism shown in Table 1-2.

Table 1-2 Mechanism to stop the WebLogic Server components

Resource Configuration	Mechanism used to stop the resource
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 operation 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 operation

The monitor operation performs the following tasks:

- Conducts a first level check on the WLS component to ensure that the WLS component's process is running. The agent identifies the process for the WLS component by applying the pattern matching described in Appendix A, "Command Line Pattern Matching for Node Manager based configurations" and Appendix B, "Command Line Pattern Matching for non-Node Manager based configurations" on page 43 on command lines of processes running in the system.
- Depending on the settings that you make, the monitor operation can conduct a second level check on the WebLogic Server component.
  The second level check uses the `wlst.cmd` scripting utility to attempt to connect to the WebLogic Server component.
  Table 1-3 lists the wlst commands used to connect to the WebLogic Server component.

Table 1-3 Commands to connect to the WLS component

Resource Configuration	Mechanism used for second level monitoring
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 operation can perform a customized check using a user-supplied monitoring utility.

Clean operation

The clean operation performs the following tasks:

- Attempts to gracefully shut down the WebLogic Server component.
- For Administrative and Managed server Node Manager based configurations, the clean operation attempts the wlst `nmKill` command.

- Identifies the process for the WLS component and kills it.

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

Chapter 2

Installing the Veritas Agent for WebLogic Server

This chapter describes the procedure to install the Veritas high availability agent for WebLogic Server. You must install the WebLogic agent on all the systems that will host a WebLogic service group.

This chapter contains the following sections:

Prerequisites

Ensure that you meet the prerequisites before installing the Veritas Agent for WebLogic Server.

- Install and configure Veritas Cluster Server.
- Remove any prior version of this agent. For details about removing an existing agent, refer to "Uninstalling the Veritas Agent for WebLogic Server" on page 33.

Upgrading the Veritas Agent for WebLogic Server

To upgrade the agent, first remove the older version of the agent. Refer to "Uninstalling the Veritas Agent for WebLogic Server" on page 33 for the uninstallation procedure. Follow the instructions given in the section "Installing the agent in a VCS environment" on page 16 to install the new agent software.

Installing the agent in a VCS environment

Use the Product Installer to install the agent for WebLogic Server.

To install the agent

1 Log into any node in the cluster as a user with domain administrative privileges.

Note: While performing installation on Windows 2003 systems, ensure that you have a user with administrative privileges logged in into the console session of each of the nodes on which you want to install the agent. To login into the console session, you can use the `mstsc /console` command.

---

2 Go to the directory mentioned in the following list:

Operating System	Directory
Windows 2000	*<cd_mount>*`/windows/w2k/application/weblogic_agent/4.3/5.0_agent/weblogic9_agt.5.0-GA_w2k`
Windows 2003	*<cd_mount>*`/windows/w2k3/application/weblogic_agent/4.3/5.0_agent/weblogic9_agt.5.0-GA_w2k3`

3 Double-click the `Setup.exe` file to begin the installation.

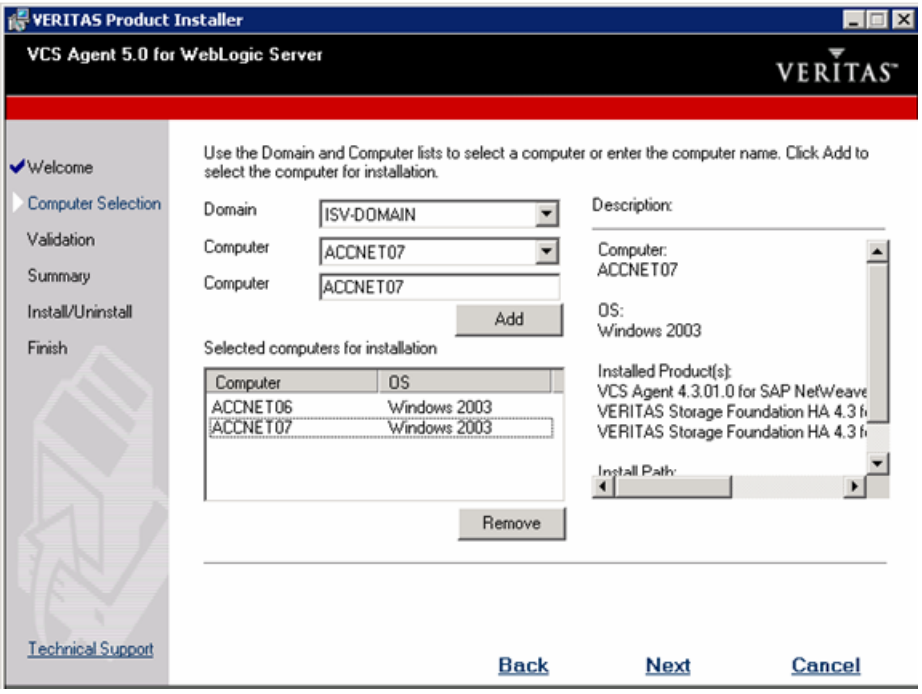
4 In the Veritas Product Installer screen, click **VCS Agent 5.0 for WebLogic Server**.

5 In the Welcome screen, click **Next**.

6 Do the following in the Computer Selection dialog box:

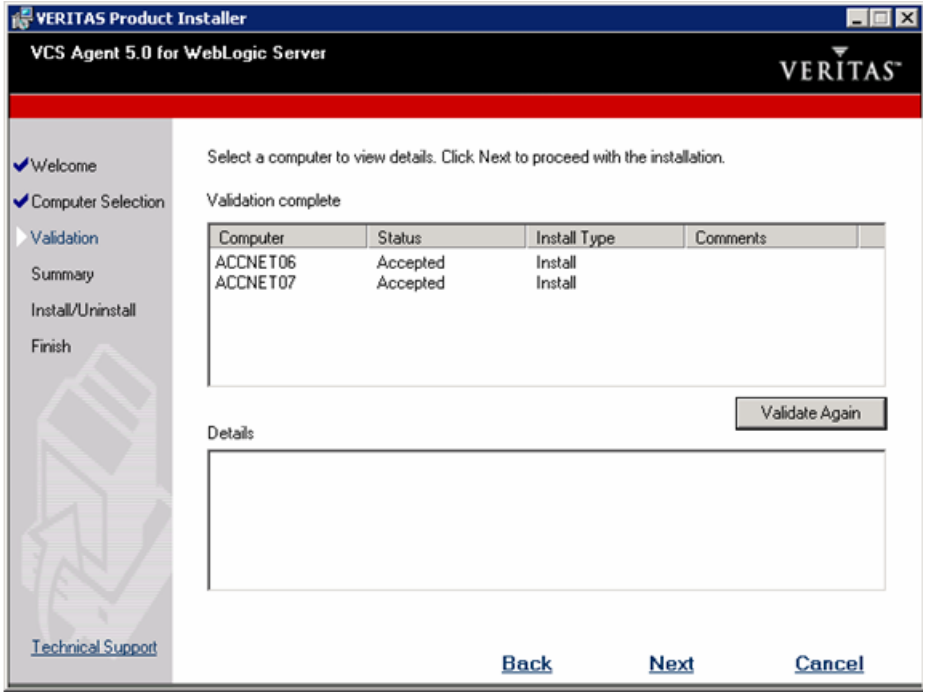
   a In the **Domain** field, select the domain in which you want to install the agent for WebLogic Server.
   The **Computer** field shows a list of computers in the domain that you selected.

b Click **Add** to select the computers in the domain on which you want to install the agent. The selected computers appear in the **Selected computers for installation** field.



c Click **Next**.

7 The Product Installer validates the installation on the selected computers and displays the status in the Validation screen. Click **Next**.



Note: If the Installer finds an error, such as the presence of a previous version of the agent on a computer, the Validation screen shows an error.

8 The Installer displays a summary report in the Report screen. Click **Install** to install the agent for WebLogic Server.
The Installer displays the installation status during and after the installation.

9 After the installation is complete, click **Next** in the Finish screen.

10 In the Veritas Product Installer screen, click **Exit** to exit the Installer.

Chapter 3

Configuring WebLogic Server High Availability using VCS 5.0 Agent for WebLogic Server

To provide high availability for WebLogic components in a WLS domain in the VCS environment, you must first configure the VCS resources. This chapter covers the steps that you must perform to configure WebLogic Server for high availability using VCS.

This chapter includes the following sections:

Configuring the Veritas Agent for WebLogic Server

After installing the agent for WebLogic Server, you can create and configure a WebLogic Server resource. Before you configure a resource, review the Agent attributes table that describes the WebLogic Server resource type and its attributes.

To view sample agent type definition file and service group, refer to ""Sample Configurations" on page 45.

The logging information generated by the agent can be seen in the agent log file, in the Cluster manager java console and in the temporary log files generated by the agent when it invokes external scripts. For more details, refer to "Inspecting agent logs" on page 37 and "Inspecting temporary log files generated by agent" on page 37.

Agent attributes

Table 3-1 describes the WebLogic Server agent attributes.

Table 3-1 Attributes

Attribute	Description
BEA_HOME String	The absolute path to the BEA Home directory of WebLogic installation. The value is used to uniquely identify the ServerRole processes. Example: `c:\bea` Default Value: No default value.
AdminURL String	The URL of the WebLogic Administrative server. The value of this attribute is used to determine if the Administrative server for the domain is fully online when the ServerRole attribute is Managed and the value of RequireAdminServer and AdminServerMaxWait attributes are set appropriately. Managed WebLogic servers use this URL to establish a connection to the Administrative server to download their configuration. This attribute is only required for a resource whose ServerRole attribute is Managed. If the ServerRole attribute is NodeManager or Administrative, no value should be specified. Example: `http://wls90host:7001` Default Value: No default value
DomainName String	The name of the WebLogic domain to which the WebLogic server belongs. The attribute is required to connect to the Node Manager using WebLogic utility wlst.cmd, which requires the DomainName and DomainDir attributes to start the Administrative and Managed servers. Example: `WLS90Domain` Default Value: No default value

Table 3-1 Attributes

Attribute	Description
DomainDir String	The domain directory for the WebLogic domain to which the WebLogic server belongs. The attribute is required to connect to the Node Manager using WebLogic utility wlst.cmd, which requires the DomainName and DomainDir attributes to start the Administrative and Managed servers. Example: `c:/bea/user_projects/domains/WLS90Domain` Default Value: No default value
ListenAddressPort String	The hostname and port of the WebLogic server. The format is hostname:port. Example: `wls90host:7001` Default Value: No default value
nmListenAddressPort String	The hostname and port of the WebLogic Node Manager. The format is hostname:port. Example: `wls90host:5556` Default Value: No default value.
nmType String	The WebLogic Node Manager type. This type is used while connecting to the Node Manager through the `wlst.cmd` script. Valid values include: ■ plain: plain socket Java-based implementation ■ rsh: RSH implementation ■ ssh: script-based SSH implementation ■ ssl: Java-based SSL implementation Example: `ssl` Default: ssl
ResLogLevel String	Specifies the logging detail performed by the agent for the resource. Valid values are: INFO - Logs error/informational messages and trace messages when error occurs. TRACE - Logs trace messages. The Trace messages are stored in the agent log when the entry point completes. To see trace messages while agent entry point is executing, add value DBG_20 to LogDbg attribute of WebLogic9 resource type. Example: `TRACE` Default: INFO
ServerName String	The name of the WebLogic server. Example: AdminServer Default: No default value

Table 3-1 Attributes

Attribute	Description
WLSUser *String*	The username used for connecting WLST to the Application Server or Node Manager. Example: `weblogic` Default: No default value
WL_HOME String	Absolute path to the Product Installation Directory of WebLogic installation. The value is used to locate wlst.cmd utility and Node Manager Home directory. Example: `c:\bea\weblogic90` Default Value: No default value.
ServerRole *String*	Type of WLS component. Must be either Administrative, Managed, or NodeManager. Example: Managed Default Value: Administrative
WLSPassword *String*	Password used for connecting WLST to Application Server or Node Manager. Encrypt the value of this attribute using the %VCS_HOME%/bin/vcsencrypt utility that VCS provides using the -**agent** option. Example: **HTIvKTlTNnINjNKnL** Default Value: No default value
AdminServerMaxWait Integer	Specifies the maximum number of seconds that a Managed server's online entry point waits for the domain's Administrative server to respond to a test probe. While this attribute is not required to successfully start, WebLogic Managed servers typically initiate a connection to the Administrative server for downloading updated configuration information. In cases in which the VCS administrator is starting all the WebLogic servers within the cluster at the same time, it would be advantageous for each Managed server to delay the start until the Administrative server has fully initialized. The AdminServerMaxWait attribute provides a way for the VCS administrator to orchestrate such a delay. The online entry point uses the AdminServerMaxWait value to control a repeating cycle of probe, wait, probe, and wait until it successfully detects the presence of the Administrative server. After it detects the server, it then proceeds with the Managed server startup. If the online entry point finds that the Administrative server is not available before the wait time expires, it generates a VCS log warning message and proceeds with server startup. Set the RequireAdminServer attribute to 1 (True) to force the online entry point to wait for a successful Administrative server response until entire duration of Online entry point. If this attribute is set to True, the online entry point ignores the value of the AdminServerMaxWait time limit. Example: 90 Default: 60

Table 3-1 Attributes

Attribute	Description
MonitorProgram String	Contains the full path name and command-line arguments for an externally provided monitor program. The monitor entry point will execute this program to perform a user defined WebLogic9 resource state check. The monitor entry point will execute the MonitorProgram under the following conditions: ■ The monitor entry point's first-level process check indicates the WebLogic9 resource is online ■ The SecondLevelMonitor is set to 0 (False) or SecondLevelMonitor is set to 1 (True) and the second-level check indicates that the WebLogic9 resource is online. This program is not supplied with the VCS agent for WebLogic and is externally developed by the end user to satisfy unique requirements. The exit code of the program will be interpreted by the monitor entry point as follows: ■ 110 or 0:The WebLogic9 resource state is ONLINE. ■ 100 or 1: The WebLogic9 resource state is OFFLINE. ■ 99: The WebLogic9 resource state is UNKNOWN. ■ Other: The WebLogic9 resource state is UNKNOWN. VERITAS recommends storing the external monitor program on the shared storage device, in the directory specified by the BEA_HOME attribute, to ensure that the file is always available on the ONLINE system. Example: `c:\bea\monitor.cmd` Default Value: No default value
RequireAdminServer Boolean	Set the RequireAdminServer attribute to 1 (True) to force the online entry point to wait for a successful Administrative server response until entire duration of Online entry point. If this attribute is set to True, the online entry point ignores the value of the AdminServerMaxWait time limit. Example: `1` (True) Default: `0` (False)

Table 3-1 Attributes

Attribute	Description
SecondLevelMonitor Integer	Used to enable second-level monitoring and specify how often it is run. Second-level monitoring is a deeper, more thorough state check of the configured WebLogic9 resource. The numeric value specifies how often that the second-level monitoring routines are run. ■ Zero (0) means never run the second-level monitoring routines. ■ One (1) would mean to run it every monitor interval. ■ Two (2) means to run the second-level monitoring routines every second monitor interval, and so on. The agent uses the BEA supplied WebLogic Scripting Tool wlst.cmd command to perform second-level monitoring using its commands `connect()`, `nmConnect()` depending upon the ServerRole. Care should be taken when setting this attribute to large numbers. For example, if the MonitorInterval is set to 60 seconds, and the SecondLevelMonitor is set to 100, then the wlst.cmd command would only get executed every 100 minutes, which may not be as often as intended. To provide maximum flexibility, the value set is not checked for an upper limit. Thus, you could cause the wlst.cmd command to run once a month, if that is what is desired. Example: 1 Default Value: 0
ServerStartProgram String	The full command line for the script used to start the WebLogic server. Example: `c:/bea/user_projects/domains/WLS90Domain/bin/startManagedWebLogic.cmd ManagedServer01 t3://wls90host:7001` Default Value: No default value
ServerStopProgram String	The full command line for the script used to stop the WebLogic server. Example: `c:/bea/user_projects/domains/WLS90Domain/bin/stopManagedWebLogic.cmd ManagedServer01 t3://wls90host:7002` Default Value: No default value

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 3-2 shows the attributes used by Node Manager based configurations.

Table 3-2 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
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
WLSPassword	-	Yes	Yes	Yes
RequireAdminServer	-	-	-	Yes
AdminServerMaxWait	-	-	-	Yes
SecondLevelMonitor	Yes	Yes	Yes	Yes
ServerStartProgram	-	-	-	-
ServerStopProgram	-	-	-	-

Table 3-3 shows the attributes used by non-Node Manager based configurations.

Table 3-3 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
DomainName	-	-	-	-
DomainDir	Opt	Opt	Opt	Opt
ListenAddressPort	-	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	Yes	Yes	Yes	Yes
ServerStartProgram	Yes	Yes	Yes	Yes
ServerStopProgram	Yes	Yes	Yes	Yes

The following list shows the kind of resource configuration and the corresponding sample configuration. You can use these sample configurations as reference while configuring your resource.

- "Node Manager without SLM enabled" on page 48.
- "Node Manager with SLM enabled" on page 49.

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.cmd 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 and stop an Administrative server instance, use the following commands:
  - To start: `<DomainDir>\bin\startWebLogic.cmd`
  - To stop: `<DomainDir>\bin\stopManagedWebLogic.cmd`

  Note: Using `stopWebLogic.cmd` to stop a Administrative server forces you to specify the user name and password in plain-text as command line parameters or as environment variables. Hence, the user can use `stopManagedWebLogic.cmd` to stop an Administrative server instance.

- To start or stop a Managed server instance, use these commands:
  - To start: `<DomainDir>\bin\startManagedWebLogic.cmd`
  - To stop: `<DomainDir>\bin\stopManagedWebLogic.cmd`

Note: Using the script `stopManagedWebLogic.cmd` to stop a Managed server with the admin_url argument causes the shutdown operation to fail when the Administrative server is unavailable. To overcome this, the user can use the `stopManagedWebLogic.cmd` with the Managed server's url as argument in place of the admin_url. Passing the Managed server's url as argument causes the script to execute WLST command `connect()` to the Managed server's URL and execute the WLST `shutdown()` command subsequently. Hence the script succeeds in shutting down the Managed server even when the Administrative server is unavailable.

Editing the WebLogic stop script

A configured resource for a WebLogic Server can use a WebLogic supplied stop script to go offline by specifying it in the ServerStopProgram attribute.

You may encounter an issue with the WebLogic supplied stop scripts, *<DomainDir>*`/bin/stopWebLogic.cmd` and *<DomainDir>*`/bin/stopManagedWebLogic.cmd`.

These stop scripts send commands to the `wlst.cmd` utility. These commands are written into a temporary file, shutdown.py.

An issue may occur if you have configured two or more VCS resources for servers belonging to the same WebLogic domain. When you attempt to bring these resources offline at the same time, all the stop scripts attempt to write the wlst commands into the same shutdown.py file. This attempt may create race conditions and some of the stop scripts may fail to complete execution.

To resolve this issue do the following:

1. Create a copy of the *<DomainDir>*`/bin/stopWebLogic.cmd` file.
2. Rename the copy as *<DomainDir>*`/bin/stopWebLogic_old.cmd`.
3. In the `stopWebLogic.cmd` file, ensure that the wlst commands are sent directly to the stdin of the wlst.cmd utility, instead of being written into a temporary file.

   For example, replace these lines:

   ```
   echo connect^(%userID% %password%
   url='%ADMIN_URL%',adminServerName='%SERVER_NAME%'^)
   >"shutdown.py" echo shutdown^('%SERVER_NAME%','Server'^)
   >>"shutdown.py" echo exit^(^) >>"shutdown.py" echo Stopping
   Weblogic Server...%JAVA_HOME%\bin\java %JAVA_OPTIONS%
   weblogic.WLST shutdown.py 2>&1
   ```

   with the following lines:

```
echo Stopping Weblogic Server...echo connect^(%userID%
%password%
url='%ADMIN_URL%',adminServerName='%SERVER_NAME%'^);shutdown^('
%SERVER_NAME%','Server'^) ;exit^(^) | %JAVA_HOME%\bin\java
%JAVA_OPTIONS% weblogic.WLST
```

Managing WebLogic servers with identical names

In a non-Node Manager based configuration, if two Administrative servers having identical names and belonging to different domains, are running on the same system, the agent monitor may yield multiple results while matching the pattern on process command lines.

To avoid any discrepancy, follow these steps for the WebLogic servers. For an Administrative server instance:

- **a** Make a copy of the startWebLogic.cmd file. Rename the copy as startWebLogic_new.cmd.
- **b** In the startWebLogic_new.cmd file, add this line:
  `set JAVA_OPTIONS=`
  `-Dweblogic.system.BootIdentityFile=<DomainDir>\servers\<Admin_server_name>\security\boot.properties`
- **c** Specify the startWebLogic_new.cmd file in the ServerStartProgram attribute for the WebLogic9 resource.
- **d** Set the value for DomainDir attribute for the WebLogic9 resource. These steps ensure that *<DomainDir>* appears in the command line for the Administrative server process. Hence, the Administrative server process is uniquely identified, even if another WebLogic Server instance with the same name is running in the system.

Avoiding storing unencrypted credentials in startup/shutdown scripts

Whenever you configure a WebLogic9 resource 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 example,
`c:\bea\wls90\admin\user_projects\domains\WLS90Domain\servers\ManagedServer01\security.`

For more details, refer to
http://edocs.bea.com/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 the cluster invokes the start or stop operation, this prompt causes the operation to fail.

Configuring WebLogic Server for high availability using VCS

Do the following steps to make WebLogic Server components that are part of a domain highly available using VCS.

1. Create and configure a VCS Service Group that consists of a Lanman, an IP address, a mount point directory, and disk group resources. Refer to the cluster documentation for details about a Service Group.
2. Bring the Service Group online.
3. Install WebLogic Server software. Ensure that you select the mount directory that you created in step 1 as the BEA home directory. While creating the domain using the config.cmd utility, ensure that you configure the WebLogic servers to listen on the virtual address of the Lanman resource.
4. Configure individual WebLogic9 resources for each of the components you want VCS to manage in the service group.
5. Attempt to:
   - Online the Service Group.
   - Offline the Service Group.
   - Switchover the Service Group to remaining systems that are part of the Service groups SystemList attribute.

Chapter 5

Uninstalling the Veritas Agent for WebLogic Server

Perform the following steps to remove the Veritas high availability agent for WebLogic Server from the cluster. You must perform these steps while the cluster is active.

To uninstall the agent

1. Ensure that all clustered VCS resources are offline.
2. From the cluster, remove all the resources that use the agent for WebLogic Server.
3. Log on to a system in the cluster from which you want to uninstall the agent as a user with administrative privileges.

Note: While performing uninstallation on Windows 2003 systems, ensure that you have a user with administrative privileges logged in into the console session of each of the nodes on which you want to uninstall the agent. To login into the console session, you can use the `mstsc /console` command.

---

4. Click **Start** > **Settings** > **Control Panel**. The Control Panel window opens.
5. Double-click **Add/Remove Programs**. The Add or Remove Programs window opens.
6. From the list of programs, select **VCS Agent 5.0 for WebLogic Server**.
7. Click **Change/Remove**.
8. Follow the instructions that the uninstall program provides, to complete the uninstallation of agent for WebLogic Server.

Chapter 6

Troubleshooting the Veritas Agent for WebLogic Server

This chapter covers issues related to VCS resources that are configured to provide high availability for WebLogic Server components in a WebLogic Server domain. The information in this chapter is intended to help you resolve issues effectively. You may come across unique issues for which you can contact Symantec support.

Using correct software and operating system versions

To ensure that no issues arise due to incorrect software and operating system versions, refer to “Supported software” on page 8 for the correct versions of operating system and software to be installed on the resource systems.

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://e-docs.bea.com/wls/docs90/secmanage/ssl.html
  - http://edocs.bea.com/wls/docs90/server_start/nodemgr.html
  - http://e-docs.bea.com/wls/docs90/secmanage/identity_trust.html

  BEA Systems 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://e-docs.bea.com/wls/docs90/ConsoleHelp/taskhelp/security/DisableHostNameVerification.html.

Unable to bring two or more VCS resources offline simultaneously

This error may occur if you have configured two or more VCS resources for servers belonging to the same WebLogic domain and VCS attempts to bring these resources offline simultaneously.

For details about resolving this issue, refer to “Editing the WebLogic stop script” on page 30.

Reviewing log files

If the configured VCS resource is not working properly, you can take a look at the log files to diagnose the problem.

Inspecting VCS log files

In case of problems while using the agent for WebLogic, you can access the VCS engine log file for more information about the particular resource.

The VCS engine log file is present in c:\program files\veritas\cluster server\log\engine_A.txt.

Inspecting agent logs

The agent log file is present in c:\program files\veritas\cluster server\log\WebLogic9_A.txt file.

Error and informational messages logged by the agent can be seen in the cluster manager java console Agent logs under WebLogic9 resource type.

Inspecting temporary log files generated by agent

The agent logs output of scripts run by it in %Windir%/temp/VRTSWebLogic9 in files of the form *<resourcename>.<entrypointname>.out*.

Example

```
c:\windows\temp\VRTSWebLogic9\wls90sg_adminserver_weblogic9.online.
out
```

Using WebLogic Server components' log files

If a WebLogic Server is facing problems, you can view the server log files to further diagnose the problem.

- For Administrative and Managed servers, the log files are located in the *<DomainDir>*\servers*<ServerName>*\logs directory.

  For example:

  ```
  c:\bea\wls90\admin\user_projects\domains\WLS90Domain\servers\ma
  nagedserver01\logs
  ```

- For Node Manager, the log files are located in the *<WL_HOME>*\common\nodemanager directory.

  For example:

  ```
  c:\bea\wls90\admin\weblogic90\common\nodemanager
  ```

Appendix A

Command Line Pattern Matching for Node Manager based configurations

This appendix contains the pattern matching that the agent applies on the command lines of processes running in the system to match the unique process for the resource.

This appendix contains the following sections:

ServerRole is NodeManager

The following pattern matching applies:

- The command line begins with `<BEA_HOME>,` followed by 0 or more characters, followed by the string `java`.
- The command line contains the string `weblogic.NodeManager`.
- The command line contains the string `ListenAddress=<nmListenAddress>` followed by a space.
- The command line contains the string `ListenPort=<nmListenPort>` followed by a space.

Example command line

```
C:\bea\JROCKI~1\jre\bin\javaw.exe -classpath
"C:\bea\JROCKI~1\jre\lib\rt.
jar;C:\bea\JROCKI~1\jre\lib\i18n.jar;C:\bea\patch_weblogic901\profi
les\default\sys_manifest_classpath\weblogic_patch.jar;C:\bea\JROCKI
~1\lib\tools.jar;C:\bea\WEBLOG~1\server\lib\weblogic_sp.jar;C:\bea\
WEBLOG~1\server\lib\weblogic.jar;C:\bea
\WEBLOG~1\server\lib\webservices.jar;C:\Program
Files\VERITAS\Security\Authentication\bin\AtWrapper.jar;C:\Program
Files\VERITAS\Security\Authentication\bin\vssatgui.jar;C:\Program
Files\VERITAS\Security\Authentication\bin\VxHelpViewer.jar;
C:\Program
Files\VERITAS\Security\Authentication\bin\VxHelpViewerl10n.jar"
-DListenAddress=localhost
-DNodeManagerHome=c:/bea/weblogic90/common/nodemanager
-DQuitEnabled=true -DListenPort=5556 weblogic.NodeManager "-v"
```

ServerRole is Administrative and ServerStartProgram is null

The following pattern matching applies:

- The command line begins with *<BEA_HOME>*, followed by 0 or more characters, followed by the string `java` followed by 0 or more characters, followed by `weblogic.Name=`*<AdminServerName>*, followed by space.
- The command line ends with `weblogic.Server`.
- The command line contains <*DomainDir*> followed by front slash or back slash.

Example command line

```
C:\bea\wls90\admin\JROCKI~1\jre\bin\java
-Dweblogic.Name=AdminServer
-Djava.security.policy=C:\bea\wls90\admin\WEBLOG~1\server\lib\weblo
gic.policy
"-Djava.library.path=C:\bea\wls90\admin\WEBLOG~1\server\bin;.;C:\WI
NDOWS\system32;C:\WINDOWS;C:\bea\wls90\admin\WEBLOG~1\server\native
\win\32;C:\bea\wls90\admin\WEBLOG~1\server\bin;C:\bea\wls90\admin\J
ROCKI~1\jre\bin;C:\bea\wls90\admin\JROCKI~1\bin;C:\bea\wls90\admin\
WEBLOG~1\server\native\win\32\oci920_8;c:\program
files\mks\mksnt;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\
Wbem;C:\Program Files\VERITAS\VERITAS Object Bus\bin;C:\Program
Files\VERITAS\VERITAS Volume Manager 4.3\;C:\Program
Files\VERITAS\VRTSjre\AccessBridge;C:\Program
Files\VERITAS\Security\Authentication\bin;C:\Program
Files\VERITAS\VRTSPerl\bin;C:\Program
Files\VERITAS\comms\llt;C:\Program
Files\VERITAS\comms\gab;C:\Program Files\VERITAS\Cluster
server\bin;C:\Program Files\VERITAS\Cluster
```

```
server\bin\VCW;C:\Program Files\Microsoft SQL Server\80\Tools\BINN"
-Djava.class.path=.;C:\bea\wls90\admin\patch_weblogic901\profiles\d
efault\sys_manifest_classpath\weblogic_patch.jar;C:\bea\wls90\admin
\JROCKI~1\lib\tools.jar;C:\bea\wls90\admin\WEBLOG~1\server\lib\webl
ogic_sp.jar;C:\bea\wls90\admin\WEBLOG~1\server\lib\weblogic.jar;C:\
bea\wls90\admin\WEBLOG~1\server\lib\webservices.jar
-Dweblogic.system.BootIdentityFile=C:\bea\wls90\admin\user_projects
\domains\WLS90Domain\servers\AdminServer\security\boot.properties
-Dweblogic.nodemanager.ServiceEnabled=true weblogic.Server
```

ServerRole is Managed and ServerStartProgram is null

The following pattern matching applies:

- Command line begins with *<BEA_HOME>*, followed by 0 or more characters, followed by the string *java*, followed by 0 or more characters, followed by `weblogic.Name=`*<ManagedServerName>*, followed by space.
- Command line ends with `weblogic.Server`.
- Command line contains *<DomainDir>* followed by front slash or back slash.
- Command line contains `management.server=`*<AdminURL>* followed by space.

Example command line

```
C:\bea\wls90\admin\JROCKI~1\jre\bin\java
-Dweblogic.Name=ManagedServer01
-Djava.security.policy=C:\bea\wls90\admin\WEBLOG~1\server\lib\weblo
gic.policy -Dweblogic.management.server=http://wls90host:7001
"-Djava.library.path=C:\bea\wls90\admin\WEBLOG~1\server\bin;.;C:\WI
NDOWS\system32;C:\WINDOWS;C:\bea\wls90\admin\WEBLOG~1\server\native
\win\32;C:\bea\wls90\admin\WEBLOG~1\server\bin;C:\bea\wls90\admin\J
ROCKI~1\jre\bin;C:\bea\wls90\admin\JROCKI~1\bin;C:\bea\wls90\admin\
WEBLOG~1\server\native\win\32\oci920_8;c:\program
files\mks\mksnt;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\
Wbem;C:\Program Files\VERITAS\VERITAS Object Bus\bin;C:\Program
Files\VERITAS\VERITAS Volume Manager 4.3\;C:\Program
Files\VERITAS\VRTSjre\AccessBridge;C:\Program
Files\VERITAS\Security\Authentication\bin;C:\Program
Files\VERITAS\VRTSPerl\bin;C:\Program
Files\VERITAS\comms\llt;C:\Program
Files\VERITAS\comms\gab;C:\Program Files\VERITAS\Cluster
server\bin;C:\Program Files\VERITAS\Cluster
server\bin\VCW;C:\Program Files\Microsoft SQL Server\80\Tools\BINN"
-Djava.class.path=.;C:\bea\wls90\admin\patch_weblogic901\profiles\d
efault\sys_manifest_classpath\weblogic_patch.jar;C:\bea\wls90\admin
```

```
\JROCKI~1\lib\tools.jar;C:\bea\wls90\admin\WEBLOG~1\server\lib\webl
ogic_sp.jar;C:\bea\wls90\admin\WEBLOG~1\server\lib\weblogic.jar;C:\
bea\wls90\admin\WEBLOG~1\server\lib\webservices.jar
-Dweblogic.system.BootIdentityFile=C:\bea\wls90\admin\user_projects
\domains\WLS90Domain\servers\ManagedServer01\data\nodemanager\boot.
properties -Dweblogic.nodemanager.ServiceEnabled=true
-Dweblogic.security.SSL.ignoreHostnameVerification=true
-Dweblogic.ReverseDNSAllowed=false weblogic.Server
```

Appendix B

Command Line Pattern Matching for non-Node Manager based configurations

This appendix contains the pattern matching that the agent applies on the command lines of processes running in the system to match the unique process for the resource.

This appendix contains the following sections:

ServerRole is Administrative and ServerStartProgram is non-null

The following pattern matching applies:

- Command line begins with `<BEA_HOME>`, followed by 0 or more characters, followed by the string `java`, followed by 0 or more characters, followed by `weblogic.Name=<AdminServerName>`, followed by space.
- Command line ends with `weblogic.Server`.
- If `<DomainDir>` is non-null, command line contains `<DomainDir>` followed by front slash or back slash.

Example command line of Admin server started with startWebLogic.cmd

```
c:\bea\wls90\admin\jdk150~1\bin\java -client   -xms256m -xmx512m
-xx:compilethreshold=8000 -xx:permsize=32m  -xx:maxpermsize=128m
-xverify:none  -da -dplatform.home=c:\bea\wls90\admin\weblog~1
-dwls.home=c:\bea\wls90\admin\weblog~1\server
-dwli.home=c:\bea\wls90\admin\weblog~1\integration
-dweblogic.management.discover=true
-dweblogic.productionmodeenabled=  -dwlw.iterativedev=
-dwlw.testconsole= -dwlw.logerrorstoconsole=
-dweblogic.ext.dirs=c:\bea\wls90\admin\patch_weblogic901\profiles\d
efault\sysext_manifest_classpath -dweblogic.Name=AdminServer
-djava.security.policy=c:\bea\wls90\admin\weblog~1\server\lib\weblo
gic.policy  weblogic.Server
```

ServerRole is Managed and ServerStartProgram is non-null

The following pattern matching applies:

- Command line begins with *<BEA_HOME>*, followed by 0 or more characters, followed by the string `java`, followed by 0 or more characters, followed by `weblogic.Name=`*<ManagedServerName>*, followed by space.
- Command line ends with `weblogic.Server`.
- If *<DomainDir>* is non-null, command line contains *<DomainDir>* followed by front slash or back slash.
- Command line contains `management.server=`*<AdminURL>* followed by space.

Example command line of server started using startManagedWebLogic.cmd

```
C:\bea\wls90\admin\JDK150~1\bin\java -server   -Xms256m -Xmx512m
-XX:MaxPermSize=128m
-Dweblogic.security.SSL.trustedCAKeyStore="C:\bea\wls90\admin\weblo
gic90\server\lib\cacerts"   -da
-Dplatform.home=C:\bea\wls90\admin\WEBLOG~1
-Dwls.home=C:\bea\wls90\admin\WEBLOG~1\server
-Dwli.home=C:\bea\wls90\admin\WEBLOG~1\integration
-Dweblogic.management.discover=false
-Dweblogic.management.server=t3://wls90host:7001
-Dwlw.iterativeDev=false -Dwlw.testConsole=false
-Dwlw.logErrorsToConsole=
-Dweblogic.ext.dirs=C:\bea\wls90\admin\patch_weblogic901\profiles\d
efault\sysext_manifest_classpath -Dweblogic.Name=ManagedServer01
Djava.security.policy=C:\bea\wls90\admin\WEBLOG~1\server\lib\weblog
ic.policy  weblogic.Server
```

Appendix C

Sample Configurations

This appendix contains samples of agent type definition, typical service group configuration, and resource configurations. The service group sample configuration graphically depicts the resource types, resources, and resource dependencies within the service group. Review these dependencies carefully before configuring the agent. For more information about these resource types, see the *Veritas Cluster Server Bundled Agents Reference Guide*.

This appendix contains the following sections:

Sample agent type definition

Following is the agent type definition.

```
type WebLogic9 (
    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
)
```

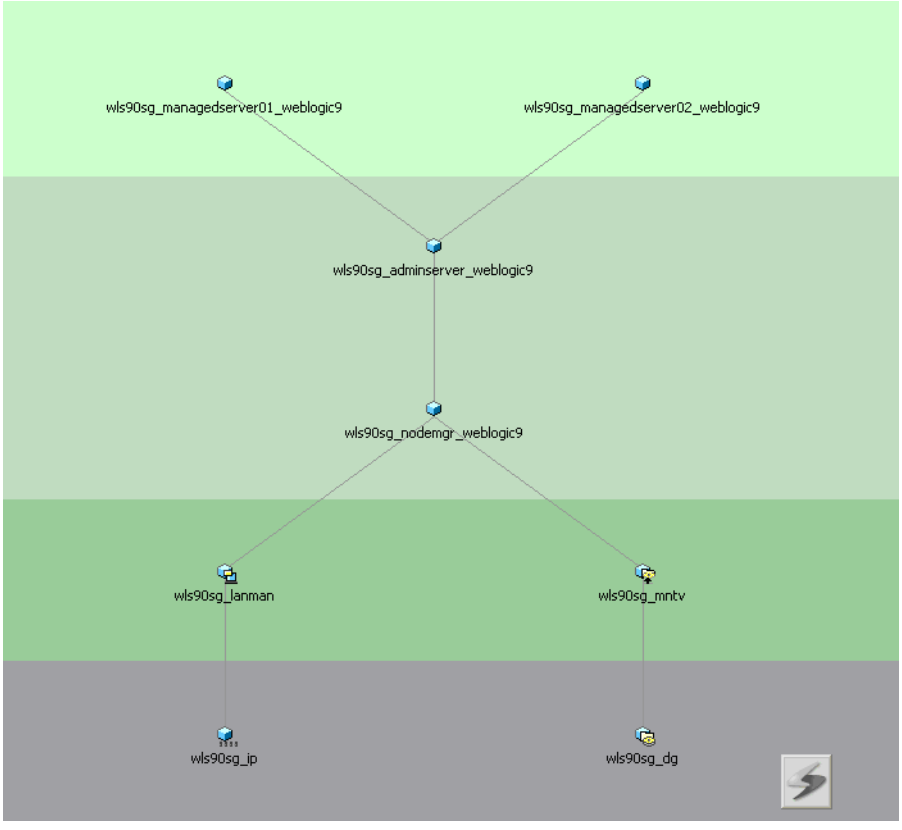
Sample service group configuration

Figure C-1 shows a sample service group configuration in a VCS environment.

In this service group configuration:

- wls90sg_nodemgr_weblogic9 is a WebLogic9 resource used to manage a WebLogic Node Manager.
- wls90sg_adminserver_weblogic9 is a WebLogic9 resource used to manage a WebLogic Administrative server.
- wls90sg_managedserver01_weblogic9 is a WebLogic9 resource used to manage a WebLogic Managed server.
- wls90sg_managedserver02_weblogic9 is a WebLogic9 resource used to manage a WebLogic Managed server.

Figure C-1 Sample service group configuration



Sample resource configurations

Figure C-2 depicts a typical configuration for Node Manager with second level monitoring (SLM) not enabled.

Figure C-2 Node Manager without SLM enabled

Figure C-3 depicts a typical configuration for Node Manager with second level monitoring (SLM) enabled.

Figure C-3 Node Manager with SLM enabled

Figure C-4 depicts a typical configuration for Administrative server (NM) with second level monitoring (SLM) not enabled.

Figure C-4 Administrative server (NM) without SLM enabled

Attribute		Value
• ResLogLevel	:	INFO
• AdminURL	:	
• BEA_HOME	:	c:\bea\wls90\admin
• WL_HOME	:	c:\bea\wls90\admin\weblogic90
• DomainName	:	wls90domain
• DomainDir	:	c:\bea\wls90\Admin\user_projects\domains\WLS90Domain
• ListenAddressPort	:	wls90host:7001
• MonitorProgram	:	
• nmListenAddressPort	:	wls90host:5558
• nmType	:	ssl
• ServerName	:	AdminServer
• ServerRole	:	Administrative
• WLSUser	:	weblogic
• WLSPassword	:	GSHuJSkSMmHMiMJmK
• RequireAdminServer	:	false
• AdminServerMaxWait	:	60
• SecondLevelMonitor	:	0
• ServerStartProgram	:	
• ServerStopProgram	:	

Figure C-5 depicts a typical configuration for Administrative server (NM) with the second level monitoring (SLM) enabled.

Figure C-5 Administrative server (NM) with SLM enabled

Attribute	Value
• ResLogLevel :	INFO
• AdminURL :	
• BEA_HOME :	c:\bea\wls90\admin
• WL_HOME :	c:\bea\wls90\admin\weblogic90
• DomainName :	wls90domain
• DomainDir :	c:\bea\wls90\Admin\user_projects\domains\WLS90Domain
• ListenAddressPort :	wls90host:7001
• MonitorProgram :	
• nmListenAddressPort :	wls90host:5558
• nmType :	ssl
• ServerName :	AdminServer
• ServerRole :	Administrative
• WLSUser :	weblogic
• WLSPassword :	GSHuJSkSMmHMiMJmK
• RequireAdminServer :	false
• AdminServerMaxWait :	60
• SecondLevelMonitor :	1
• ServerStartProgram :	
• ServerStopProgram :	

Figure C-6 depicts a typical configuration for Managed server (NM) with second level monitoring (SLM) not enabled.

Figure C-6 Managed server (NM) without SLM enabled

Attribute		Value
• ResLogLevel	:	INFO
• AdminURL	:	http://wls90host:7001
• BEA_HOME	:	c:\bea\wls90\admin
• WL_HOME	:	c:\bea\wls90\admin\weblogic90
• DomainName	:	WLS90Domain
• DomainDir	:	c:\bea\wls90\Admin\user_projects\domains\WLS90Domain
• ListenAddressPort	:	wls90host:7002
• MonitorProgram	:	
• nmListenAddressPort	:	wls90host:5558
• nmType	:	ssl
• ServerName	:	ManagedServer01
• ServerRole	:	Managed
• WLSUser	:	weblogic
• WLSPassword	:	IUJwLUmUOoJOkOLoM
• RequireAdminServer	:	false
• AdminServerMaxWait	:	10
• SecondLevelMonitor	:	0
• ServerStartProgram	:	
• ServerStopProgram	:	

Figure C-7 depicts a typical configuration for Managed server (NM) with second level monitoring (SLM) enabled.

Figure C-7 Managed server (NM) with SLM enabled

Attribute		Value
• ResLogLevel	:	INFO
• AdminURL	:	http://wls90host:7001
• BEA_HOME	:	c:\bea\wls90\admin
• WL_HOME	:	c:\bea\wls90\admin\weblogic90
• DomainName	:	WLS90Domain
• DomainDir	:	c:\bea\wls90\Admin\user_projects\domains\WLS90Domain
• ListenAddressPort	:	wls90host:7002
• MonitorProgram	:	
• nmListenAddressPort	:	wls90host:5558
• nmType	:	ssl
• ServerName	:	ManagedServer01
• ServerRole	:	Managed
• WLSUser	:	weblogic
• WLSPassword	:	IUJwLUmUOoJOkOLoM
• RequireAdminServer	:	false
• AdminServerMaxWait	:	10
• SecondLevelMonitor	:	1
• ServerStartProgram	:	
• ServerStopProgram	:	

Figure C-8 depicts a typical configuration for Managed server (NNM) with the second level monitoring (SLM) not enabled.

Figure C-8 Managed server (NNM) without SLM enabled

Attribute	Value
ResLogLevel	INFO
AdminURL	t3://wls90host:7001
BEA_HOME	c:\bea\wls90\admin
WL_HOME	
DomainName	
DomainDir	
ListenAddressPort	
MonitorProgram	
nmListenAddressPort	
nmType	
ServerName	ManagedServer02
ServerRole	Managed
WLSUser	
WLSPassword	
RequireAdminServer	false
AdminServerMaxWait	0
SecondLevelMonitor	0
ServerStartProgram	c:\bea\wls90\admin\user_projects\domains\wls90domain\bin\startManagedWebLogic.cmd ManagedServer02 t3://wls90host:7001
ServerStopProgram	c:\bea\wls90\admin\user_projects\domains\wls90domain\bin\stopManagedWebLogic.cmd ManagedServer02 t3://wls90host:7004

Figure C-9 depicts a typical configuration for Managed server (NNM) with second level monitoring (SLM) enabled.

Figure C-9 Managed server (NNM) with SLM enabled

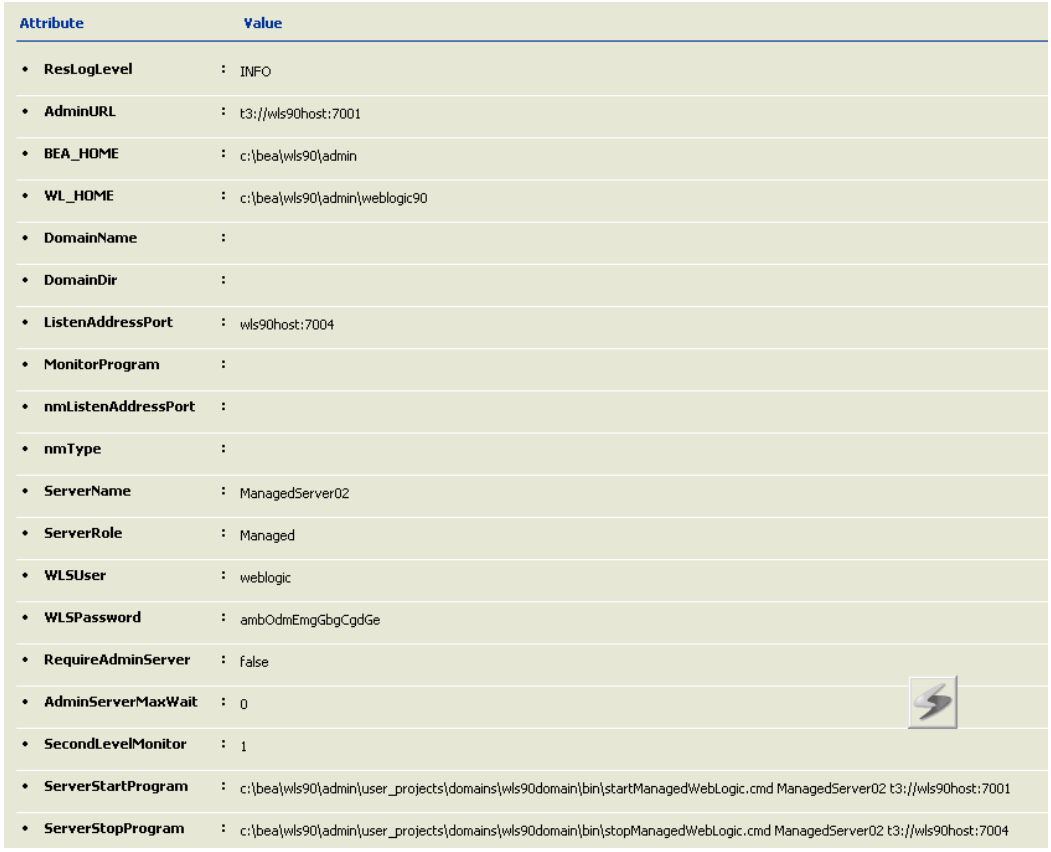


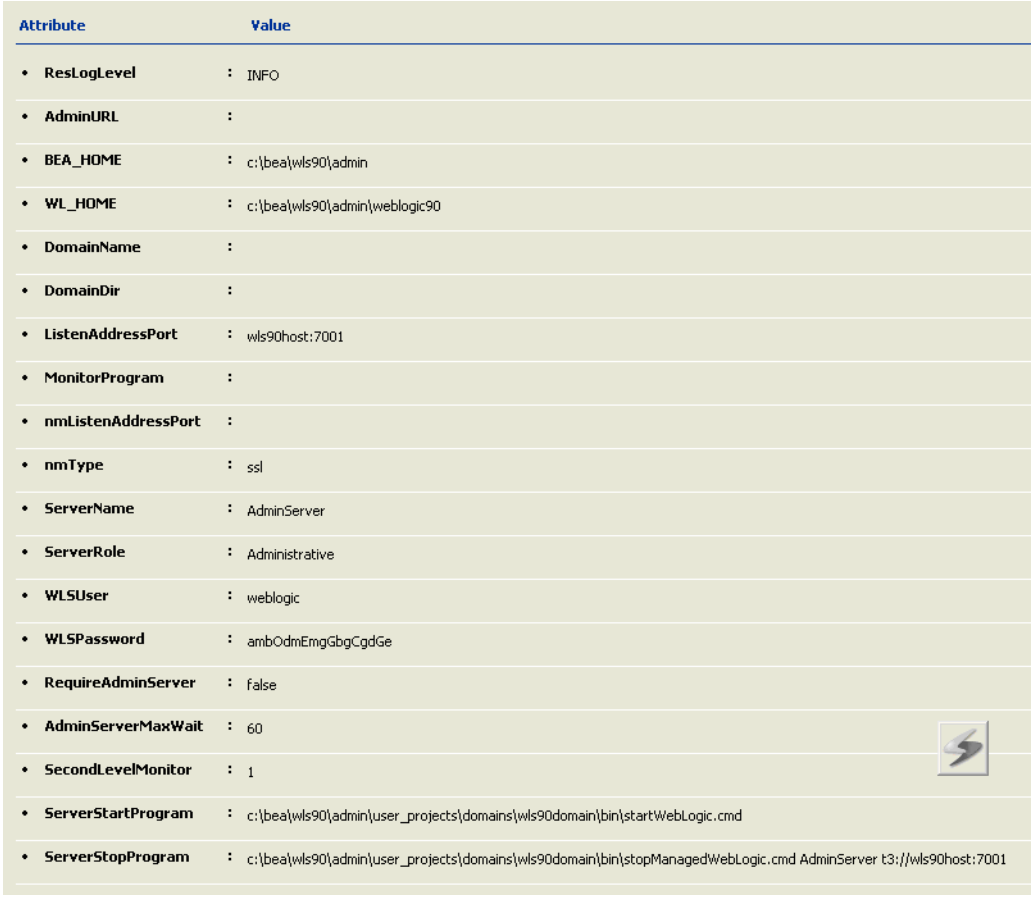
Figure C-10 depicts a typical configuration for Administrative server (NNM) with second level monitoring (SLM) not enabled.

Figure C-10 Administrative server (NNM) without SLM enabled

Attribute	Value
ResLogLevel	INFO
AdminURL	
BEA_HOME	c:\bea\wls90\admin
WL_HOME	
DomainName	
DomainDir	
ListenAddressPort	
MonitorProgram	
nmListenAddressPort	
nmType	ssl
ServerName	AdminServer
ServerRole	Administrative
WLSUser	
WLSPassword	
RequireAdminServer	false
AdminServerMaxWait	60
SecondLevelMonitor	0
ServerStartProgram	c:\bea\wls90\admin\user_projects\domains\wls90domain\bin\startWebLogic.cmd
ServerStopProgram	c:\bea\wls90\admin\user_projects\domains\wls90domain\bin\stopManagedWebLogic.cmd AdminServer t3://wls90host:7001

Figure C-11 depicts a typical configuration for Administrative server (NNM) with the second level monitoring (SLM) enabled.

Figure C-11 Administrative server (NNM) with SLM enabled



Index