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

AIX, HP-UX, Linux, Solaris

5.1

✓ Symantec™

Veritas High Availability Agent for WebLogic Server Installation and Configuration Guide

The software described in this book is furnished under a license agreement and may be used only in accordance with the terms of the agreement.

Agent Version: 5.1.13.0

Document version: 5.1.13.0.0

Legal Notice

Technical Support

Symantec Technical Support maintains support centers globally. Technical Support's primary role is to respond to specific queries about product features and functionality. The Technical Support group also creates content for our online Knowledge Base. The Technical Support group works collaboratively with the other functional areas within Symantec to answer your questions in a timely fashion. For example, the Technical Support group works with Product Engineering and Symantec Security Response to provide alerting services and virus definition updates.

Symantec's support offerings include the following:

■ A range of support options that give you the flexibility to select the right amount of service for any size organization

■ Telephone and/or Web-based support that provides rapid response and up-to-the-minute information

■ Upgrade assurance that delivers software upgrades

■ Global support purchased on a regional business hours or 24 hours a day, 7 days a week basis

■ Premium service offerings that include Account Management Services

For information about Symantec's support offerings, you can visit our Web site at the following URL:

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

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

Contacting Technical Support

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

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

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

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

■ Product release level

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

Licensing and registration

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

www.symantec.com/business/support/

Customer service

Customer service information is available at the following URL:

www.symantec.com/business/support/

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

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

Support agreement resources

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

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

Documentation

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

https://sort.symantec.com/documents

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

doc_feedback@symantec.com

About Symantec Connect

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

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

Contents

Introducing the Veritas High Availability Agent for WebLogic Server

This chapter includes the following topics:

- About the Veritas agent for WebLogic Server

- What's new in this agent

- Supported software

- About WebLogic Server

- WebLogic Server agent functions

About the Veritas agent for WebLogic Server

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

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

See the following Technical Support TechNote for the latest updates or software issues for this agent:

http://www.symantec.com/business/support/index?page=content&id=TECH46455

What's new in this agent

The enhancement in this release of the WebLogic Server agent is as follows:

■ Added Intelligent Monitoring Framework (IMF) capability and support for intelligent resource monitoring.

■ The SecondLevelMonitor attribute of the Veritas agent for WebLogic Server is deprecated if both the following software versions are installed:

■ Veritas Cluster Server (VCS) 5.1 SP1

■ Veritas agent for WebLogic Server version 5.1.13.0

In such a case, instead of the SecondLevelMonitor attribute, you must use the resource type level attribute—LevelTwoMonitorFreq—to specify the frequency of in-depth monitoring. For more information: See Table 3-2 on page 35.

■ Added support for AIX 7.1.

■ Changed the process filter such that during the monitor function, the agent now excludes the protocol from the value of the AdminURL attribute. During the monitor function for the WebLogic managed server, the agent matches the ListenAddress:Port value from the AdminURL attribute to the management.server Java argument of the managed server process.

For information on the changes introduced in the previous releases:

See "Changes introduced in previous releases" on page 97.

Supported software

The Veritas agent for Weblogic Server supports the following software versions in a VCS environment.

Veritas Cluster Server	■ AIX—VCS 4.0, 5.0, 5.1
	■ HP-UX—VCS 4.1, 5.0
	■ Linux—VCS 4.0, 4.1, 5.0, 5.1
	■ Solaris—VCS 4.0, 4.1, 5.0, 5.1
	and all intermediate Maintenance Packs of these releases.
ACC Library	5.1 and later
	Review the ACC Library version for i18n support.
	See "Prerequisites for enabling i18n support" on page 20.

Operating Systems	■ AIX 5.1, 5.2, 5.3, 6.1, 7.1 on pSeries
	■ HP-UX 11i v2, 11i v3 on Itanium and PA-RISC
	■ Red Hat Enterprise Linux 3.0, 4.0, 5.0 on Intel
	■ SUSE Linux Enterprise Server 9, 10
	■ Solaris 8, 9, 10 on SPARC and x86 including the zones and logical domains.
WebLogic Server	7.0, 8.1, 9.0, 9.1, 9.2, 10.0, 10.3(10gR3), 10.3.1(11gR1), and 10.3.x (such as 10.3.1 and 10.3.2).

Support matrix for IMF and in-depth monitoring

Depending on your version of Veritas Cluster Server (VCS) and the WebLogic Server agent, the following features and functionality are supported.

Table 1-1   IMF and detail monitoring support matrix

VCS and agent version	IMF capability	SecondLevelMonitor attribute	LevelTwoMonitorFreq attribute
VCS 5.1 SP1 or later with WebLogic Server agent 5.1.13.0 or later	Yes	No	Yes
VCS 5.1 SP1 with WebLogic Server agent 5.1.12.0 or earlier	No	Yes	No
VCS 5.1 or earlier with WebLogic Server agent 5.1.13.0 or later	No	Yes	No
VCS 5.1 or earlier with WebLogic Server agent 5.1.12.0 or earlier	No	Yes	No

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 BEA WebLogic server and BEA 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.
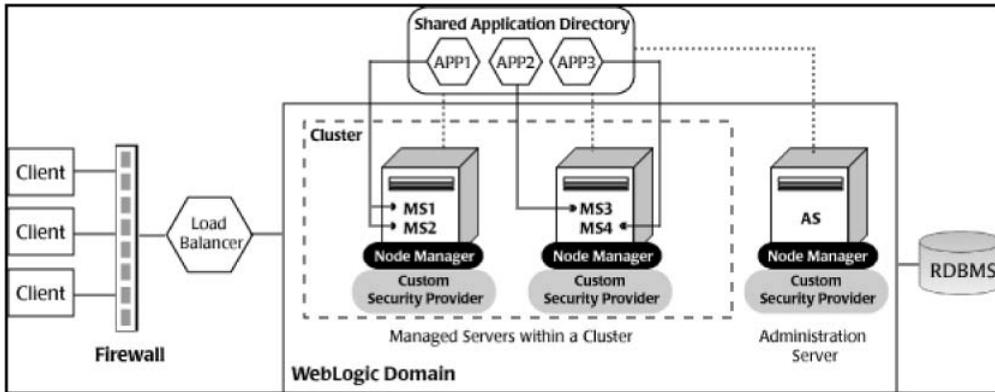
The Veritas agent for WebLogic Server supports both Administrative and Managed Servers, and Node Manager based configurations. The agent recognizes the startup server dependency that exists between Managed and Administrative Servers and provides the cluster administrator with the choice of enforcing or not enforcing this startup restriction. Similarly, the agent is WebLogic Cluster agnostic. In other words, this agent can provide clustering services for stand-alone WebLogic Servers and can support Managed Servers that participate in a WebLogic Cluster.

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

The online function 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 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

The offline function performs the following 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.

  The agent for WebLogic Server also supports Intelligent Monitoring Framework (IMF) in the first-level check. IMF enables intelligent resource monitoring. The agent for WebLogic Server is IMF-aware and uses the asynchronous monitoring framework (AMF) kernel driver for resource state change notifications. See "How the agent supports intelligent resource monitoring" on page 50.

  You can use the MonitorFreq key of the IMF attribute to specify the frequency at which the agent invokes the monitor function. See "MonitorFreq" on page 52.

- 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`.

Note: The attribute used to configure the second-level check and its frequency depends on the software versions of VCS and WebLogic Server agent you have installed: For VCS 5.1 SP1 or later with WebLogic Server agent version 5.1.13.0, use the LevelTwoMonitorFreq attribute. For VCS 5.1 or earlier with WebLogic Server agent 5.1.12.0 or earlier, use the SecondLevelMonitor attribute.

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

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.

Note: For information about the additional functions of the agent for WebLogic Server when IMF is enabled: See "Agent functions for the IMF functionality" on page 51.

Installing, upgrading, and removing the agent for WebLogic Server

This chapter includes the following topics:

Before you install the Veritas agent for WebLogic Server

You must install the Veritas agent for WebLogic Server on all the systems that will host WebLogic Server service groups.

Ensure that you meet the following prerequisites to install the agent for WebLogic Server.

For VCS, do the following:

- Install and configure Veritas Cluster Server.

For more information on installing and configuring Veritas Cluster Server, refer to the *Veritas Cluster Server Installation Guide*.

■ If the operating system is HP-UX 11i v1, install patch PHCO_29042.

■ Remove any previous version of this agent.
  To remove the agent,
  See "Removing the agent in a VCS environment" on page 23.

■ Install the latest version of ACC Library.
  To install or update the ACC Library package, locate the library and related documentation in the agentpack tarball:
  See "Installing the ACC library" on page 21.

Prerequisites for enabling i18n support

Perform the following steps to enable i18n support to the agent:

■ Install ACCLib version 5.1.2.0 or later.
  See "Installing the ACC library" on page 21.

■ For VCS 5.0 and earlier releases, copy the latest ag_i18n_inc.pm module from the following location on the agent pack disc.

Note: Review the readme.txt for instructions to copy this module.

VCS 5.0	cd1/*platform*/*arch_dist*/vcs/application/i18n_support/5.0
VCS 4.1	cd1/*platform*/*arch_dist*/vcs/application/i18n_support/4.1
VCS 4.0	cd1/*platform*/*arch_dist*/vcs/application/i18n_support/4.0

where *arch_dist* takes the following values:
'sol_sparc' for Solaris SPARC
'sol_x64' for Solaris x64
'generic' for HP-UX and Linux

Note: *arch_dist* is not applicable to AIX.

About the ACC library

The operations of a VCS agent depend on a set of Perl modules known as the ACC library. The library must be installed on each system in the cluster that runs the

agent. The ACC library contains common, reusable functions that perform tasks, such as process identification, logging, and system calls.

Instructions to install or remove the ACC library on a single system in the cluster are given in the following sections. The instructions assume that the agent's tar file has already been extracted.

Installing the ACC library

Install the ACC library on each system in the cluster that runs an agent that depends on the ACC library.

To install the ACC library

1   Log in as superuser.

2   Download ACC Library.

   You can download either the complete Agent Pack tarball or the individual ACCLib tarball from the Symantec Operations Readiness Tools (SORT) site (https://sort.symantec.com/agents).

3   If you downloaded the complete Agent Pack tarball, navigate to the directory containing the package for the platform running in your environment.

AIX	*cd1*/aix/vcs/application/acc_library/*version*_library/pkgs
HP-UX	*cd1*/hpux/generic/vcs/application/acc_library/*version*_library/pkgs
Linux	*cd1*/linux/generic/vcs/application/acc_library/*version*_library/rpms
Solaris	*cd1*/solaris/*dist_arch*/vcs/application/acc_library/*version*_library/pkgs
	where *dist_arch* is sol_sparc or sol_x64.

4   If you downloaded the individual ACCLib tarball, navigate to the pkgs directory (for AIX, HP-UX, and Solaris), or rpms directory (for Linux).

5   Install the package. Enter **Yes** if asked to confirm overwriting of files in the existing package.

AIX	`# installp -ac -d VRTSacclib.bff VRTSacclib`
HP-UX	`# swinstall -s 'pwd' VRTSacclib`
Linux	`# rpm -i \` `VRTSacclib-VersionNumber-GA_GENERIC.noarch.rpm`
Solaris	`# pkgadd -d VRTSacclib.pkg`

Installing the agent in a VCS environment

Install the agent for WebLogic Server on each node in the cluster.

To install the agent in a VCS environment

1   Download the agent from the Symantec Operations Readiness Tools (SORT) site: https://sort.symantec.com/agents.

    You can download either the complete Agent Pack tarball or an individual agent tarball.

2   Uncompress the file to a temporary location, say /tmp.

3   If you downloaded the complete Agent Pack tarball, navigate to the directory containing the package for the platform running in your environment.

AIX	`cd1/aix/vcs/application/weblogic_agent/` *`vcs_version`*`/`*`version`*`_agent/pkgs`
HP-UX	`cd1/hpux/generic/vcs/application/weblogic_agent/` *`vcs_version`*`/`*`version`*`_agent/pkgs`
Linux	`cd1/linux/generic/vcs/application/weblogic_agent/` *`vcs_version`*`/`*`version`*`_agent/rpms`
Solaris	`cd1/solaris/`*`dist_arch`*`/vcs/application/weblogic_agent/` *`vcs_version`*`/`*`version`*`_agent/pkgs`
	where, *dist_arch* is sol_x64 or sol_sparc

    If you downloaded the individual agent tarball, navigate to the pkgs directory (for AIX, HP-UX, and Solaris), or rpms directory (for Linux).

4   Log in as superuser.

5   Install the package.

AIX   # installp -ac -d VRTSwls.rte.bff VRTSwls.rte

HP-UX   # swinstall -s 'pwd' VRTSwls

Linux   # rpm -ihv \
          VRTSwls-*AgentVersion*-GA_GENERIC.noarch.rpm

Solaris   # pkgadd -d . VRTSwls

6   After installing the agent package, you must import the agent type
    configuration file. See "Importing the agent types files in a VCS environment"
    on page 28.

Removing the agent in a VCS environment

You must uninstall the agent for WebLogic Server from a cluster while the cluster
is active.

To uninstall the agent in a VCS environment

1   Log in as a superuser.

2   Set the cluster configuration mode to read/write by typing the following
    command from any node in the cluster:

    # haconf -makerw

3   Remove all WebLogic Server resources from the cluster. Use the following
    command to verify that all resources have been removed:

    # hares -list Type=WebLogic

4   Remove the agent type from the cluster configuration by typing the following
    command from any node in the cluster:

    # hatype -delete WebLogic

    Removing the agent's type file from the cluster removes the include statement
    for the agent from the main.cf file, but the agent's type file is not removed
    from the cluster configuration directory. You can remove the agent's type
    file later from the cluster configuration directory.

5 Save these changes. Then set the cluster configuration mode to read-only by typing the following command from any node in the cluster:

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

6 Use the platform's native software management program to remove the agent for WebLogic Server from each node in the cluster.

Execute the following command to uninstall the agent:

AIX	# `installp -u VRTSwls.rte`
HP-UX	# `swremove VRTSwls`
Linux	# `rpm -e VRTSwls`
Solaris	# `pkgrm VRTSwls`

Removing the ACC library

Perform the following steps to remove the ACC library.

To remove the ACC library

1 Ensure that all agents that use ACC library are removed.

2 Run the following command to remove the ACC library package.

AIX	# `installp -u VRTSacclib`
HP-UX	# `swremove VRTSacclib`
Linux	# `rpm -e VRTSacclib`
Solaris	# `pkgrm VRTSacclib`

Upgrading the agent in a VCS environment

Perform the following steps to upgrade the agent with minimal disruption, in a VCS environment.

Updating the agent using the update script

The WebLogic package (VRTSwls) contains a script named `wls_update.pl`. This script updates the WebLogic9 agent type to WebLogic and reconfigures all the existing resources of the WebLogic9 agent type with the WebLogic agent type. This script also removes the WebLogic9 agent type.

To update the agent using the update script

1 Install the WebLogic Server agent package (VRTSwls) on all nodes.

See "Installing the agent in a VCS environment" on page 22.

2 Navigate to the agent source directory:

3 Execute the wls_update.pl script available in the agent directory.

```
bash-3.00# ./wls_update.pl
```

4 Follow the prompts to update the agent type.

5
```
Do you want to go ahead with Update process (y/n):y

Are you sure to update [WebLogic9] Type to [WebLogic] Type (y/n):y

Do you want to Delete the Agent Type WebLogic9(y/n):y
```

6 If the script runs successfully, the following messages are displayed:

```
Old Agent type [WebLogic9]  has been deleted

The old agent type[ WebLogic9 ] is successfully replaced
by new agent type [WebLogic]

For more details check the log file
[/var/VRTSvcs/log/tmp/WebLogic_10881/wls_update.log]
```

7 Use the platform's native software management program to remove the old agent package (VRTSwls9) for WebLogic Server from each node in the cluster.

Execute the following command to uninstall the agent:

AIX	# installp -u VRTSwls9.rte
HP-UX	# swremove VRTSwls9
Linux	# rpm -e VRTSwls9
Solaris	# pkgrm VRTSwls9

8 This completes the procedure for updating the WebLogic9 agent type to WebLogic.

Note: If you upgraded the VCS version to VCS 5.1 SP1 and the WebLogic Server agent version to 5.1.13.0 (or later), and if you had enabled detail monitoring in the previous version, then do the following: Set the value of the LevelTwoMonitorFreq attribute to the same value as that of the SecondLevelMonitor attribute.

Configuring the agent for WebLogic Server

This chapter includes the following topics:

About configuring the Veritas agent for WebLogic Server

After installing the Veritas agent for WebLogic Server, you must import the agent type configuration file. After importing this file, you can create and configure

WebLogic Server resources. Before you configure a resource, review the attributes table that describes the resource type and its attributes.

To view the sample agent type definition and service groups configuration.

See "About sample configurations for the agent for WebLogic Server" on page 77.

Importing the agent types files in a VCS environment

To use the agent for WebLogic Server, you must import the agent types file into the cluster.

To import the agent types file using the Veritas Cluster Server graphical user interface

1 Start the Veritas Cluster Manager and connect to the cluster on which the agent is installed.

2 Click **File > Import Types**.

3 In the Import Types dialog box, select the following file:

VCS version	Operating System	Agent types file
VCS 4.x	■ AIX ■ HP-UX ■ Linux ■ Solaris	/etc/VRTSvcs/conf/sample_WebLogic/ WebLogicTypes.cf
VCS 5.x	■ AIX ■ HP-UX ■ Linux	/etc/VRTSagents/ha/conf/WebLogic/ WebLogicTypes.cf
VCS 5.0	■ Solaris SPARC and x64	/etc/VRTSagents/ha/conf/WebLogic/ WebLogicTypes50.cf
VCS 5.1	■ Solaris SPARC and x64	/etc/VRTSagents/ha/conf/WebLogic/ WebLogicTypes51.cf

4 Click **Import**.

5 Save the VCS configuration.

The WebLogic Server agent type is now imported to the VCS engine.

You can now create WebLogic Server resources. For additional information about using the VCS GUI, refer to the *Veritas Cluster Server User's Guide*.

To import the agent types file using the Veritas Cluster Server command line interface (CLI), perform the following steps.

1 Log on to any one of the systems in the cluster as the superuser.

2 Create a temporary directory.

```
# mkdir ./temp
```

```
# cd ./temp
```

3 Copy the sample file Types.cf.

4 Create a dummy main.cf file:

```
# echo 'include "WebLogicTypes.cf"' > main.cf
```

5 Create the WebLogic Server resource type as follows:

```
# hacf -verify .
```

```
# haconf -makerw
```

```
# sh main.cmd
```

```
# haconf -dump
```

The WebLogic Server agent type is now imported to the VCS engine.

You can now create WebLogic Server resources. For additional information about using the VCS CLI, refer to the *Veritas Cluster Server User's Guide*.

WebLogic Server agent attributes

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

Table 3-1 lists the required attributes for the agent for WebLogic Server.

Table 3-1   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 3-1	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
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
nmHome	The absolute path to the Node Manager home directory.
	Specify this attribute for Node Manager based configurations only.
	Type and dimension: string-scalar
	Default: ""
	Example 1: /bea/wls90/admin/weblogic90/common/nodemanager
	Example 2: /bea/wls90/admin/user_projects/domains/nodemanager
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

Table 3-1   Required attributes *(continued)*

Required attribute	Description
nmUser	The user name of the Node Manager. The agent uses this user identity, along with the nmPassword, to connect to the Node Manager. Specify this attribute for Node Manager based configurations only. Type and dimension: string-scalar Example: weblogic Default: ""
nmPassword	The password of the Node Manager. The agent uses this password, along with the nmUser, to connect to the Node Manager. For VCS, encrypt the value of this attribute using the $VCS_HOME/bin/vcsencrypt utility that VCS provides. Specify this attribute for Node Manager based configurations only. Type and dimension: string-scalar Default: "" Example: HTIvKTlTNnINjNKnL
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 3-1	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 40.
	Type and dimension: string-scalar
	Default: ""
	Example: AdminServer
ShutdownTimeout	The timeout value, in milliseconds, which is passed on to the wlst.sh script. The wlst.sh script uses this value to instruct the agent to shut down a specified server forcefully, if the shutdown() command is not completed in the given timeframe.
	■ If the value of this attribute is set to 0, the agent tries to gracefully shut down the specified server till the value specified in the OfflineTimeout attribute of the Offline function. If even after exceeding the value specified in the OfflineTimeout attribute, the resource is still not offline, the agent runs the Clean function. ■ If the value of this attribute is greater than 0, then that value is passed on to shutdown() command via the wlst.sh script.
	Type and dimension: integer-scalar
	Default: ""
	Example: 1000
WlstScript	The absolute path to the wlst.sh utility.
	Type and dimension: string-scalar
	Default: ""
	Example: /bea/wls90/admin/weblogic90/common/bin/wlst.sh
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 3-1   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
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: ■ For Administrative Server ServerStartProgram = "/wls/my_domain/startWebLogic.sh" ■ For Managed Server ServerStartProgram = "/wls/my_domain/startManagedWebLogic.sh Managed1 t3://wlsadmin:7001" ServerStartProgram = "/wls/my_domain/startManagedWebLogic.sh Managed1 t3s://wlsadmin:9002" If no arguments are passed, 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/startWebLogic.sh

	Table 3-1	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:
	■ For Administrative Server ServerStopProgram = "/wls/my_domain/stopWebLogic.sh" ServerStopProgram = "/wls/my_domain/stopWebLogic.sh wlsuser wlspasswd t3s://wlsadmin:9002" ■ For Managed Server ServerStopProgram = "/wls/my_domain/stopManagedWebLogic.sh Managed1 t3://wlsadmin:7001 wlsuser wlspasswd" ServerStopProgram = "/wls/my_domain/stopManagedWebLogic.sh Managed1 t3s://wlsadmin:9002 wlsuser wlspasswd"
	If no arguments are passed, the agent forms the command line as follows:
	■ For Managed Server: `$ServerStopProgram $ServerName $AdminURL $WLSUser $WLSPassword` ■ For Administrative Server: `$ServerStopProgram`
	Type and dimension: string-scalar
	Default: ""
	Example: /bea/user_projects/domains/WLS90Domain/bin/stopWebLogic.sh
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 3-1   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 $VCS_HOME/bin/vcsencrypt utility that VCS provides. While encrypting the password, use the -agent option. Type and dimension: string-scalar Default: "" Example: weblogic

Table 3-2 lists the optional attributes.

Table 3-2   Optional attributes

Optional attribute	Description
AdminUrl	The URL of the Administrative Server. ■ If the Administrative Server uses http and t3 protocols, set this attribute only for those resources whose ServerRole attribute is Managed. ■ If the Administrative server uses https and t3s protocols, set this attribute for those resources whose ServerRole attribute is Administrative or Managed. For the Managed Server, ensure that the value of this attribute contains the Administrative Server host name and port that appear in the long listing of processes for the Managed Server. If the value of this attribute is different, the agent fails to find the process. For example, the value of this attribute could be *<protocol>*://wlsadmin:7001, where wlsadmin is the Administrative Server host name, and 7001 is the Administrative Server port. If the RequireAdminServer attribute is set to 1, AdminUrl is used to connect to the Administrative Server of the specified 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 1: http://wlsadmin:7001 Example 2: https://wlsadmin:7001

| | **Table 3-2** | 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 46. 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 39. 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 3-2 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.
	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.
	For information on how the agent executes second-level monitoring depending on the version of WebLogic Server installed: See "Executing second-level monitoring" on page 38.
	Note: The SecondLevelMonitor attribute is applicable to VCS versions earlier than VCS 5.1 SP1 with WebLogic Server agent versions earlier than 5.1.13.0. From VCS version 5.1 SP1 with WebLogic Server agent version 5.1.13.0 onwards, the SecondLevelMonitor attribute is deprecated. Instead, a resource type level attribute LevelTwoMonitorFreq should be used to specify the frequency of in-depth monitoring.
	Type and dimension: integer-scalar
	Default: 0
	Example: 1

Table 3-2   Optional attributes *(continued)*

Optional attribute	Description
LevelTwoMonitorFreq	Specifies the frequency at which the agent for this resource type must perform second-level or detailed monitoring. You can also override the value of this attribute at the resource level. The value indicates the number of monitor cycles after which the agent will monitor the WebLogic server in detail.
	For example, the value 5 indicates that the agent will monitor the WebLogic server in detail after every five online monitor intervals.
	Note: This attribute is applicable to VCS version 5.1 SP1 or later with WebLogic Server agent version 5.1.13.0 or later. If the VCS version is earlier than VCS 5.1 SP1 and the WebLogic Server agent version is earlier than 5.1.13.0, the SecondLevelMonitor attribute should be used.
	If you upgraded the VCS version to VCS 5.1 SP1 or later and the WebLogic Server agent version to 5.1.13.0 (or later), and if you had enabled detail monitoring in the previous version, then do the following:
	■ Set the value of the LevelTwoMonitorFreq attribute to the same value as that of the SecondLevelMonitor attribute.
	Type and dimension: integer-scalar
	Default: 0

Note: For information about the additional attributes of the agent for WebLogic Server when IMF is enabled: See "Attributes that enable IMF" on page 51.

Executing second-level monitoring

The procedure for WebLogic Server version 7.0 and 8.1 is as follows:

The monitor function performs tests as part of this second-level state check, depending on the agent configuration "weblogic.Admin GETSTATE" Test. BEA bundles a command-line administration utility called weblogic.Admin with every WebLogic Server distribution. This utility provides a variety of administrative functions allowing one to fully control a WebLogic Server without the use of the Administrative Server Web Console or the WebLogic Administrative Portal. The "GETSTATE" option of the weblogic.Admin command-line interface establishes a connection to the target server, probes the server and returns server state information. This is generally considered to be the most robust and reliable WLS state probe.

To run the weblogic.Admin command, security credentials need to be stored so that the command can be run in a background mode without user interaction.

These credentials are stored in two files created using the STOREUSERCONFIG command option. The credential files can be arbitrarily named so that the monitor agent looks for the files named VRTSWebLogicKey.properties and VRTSWebLogicConfig.properties in the DomainDir directory. If these files exist, the monitor function uses the weblogic.Admin command to probe the WebLogic Server state.

See "Configuring "weblogic.Admin GETSTATE" based monitoring" on page 77.

The procedure for WebLogic Server version 9.0, 9.1, 9.2, and 10.0 is as follows:

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.

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. Using shared disk and virtual IP addresses, you can manage a large set of WebLogic Server instances in a single cluster.

WebLogic Servers can run on separate cluster nodes or can run concurrently on a single node. In the later case, 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 DomainDir must be unique for the Administrative Server instance.

- To uniquely identify a Managed Server instance, do the following:

    - The combination of ServerName and DomainDir 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 3-3 shows the attributes used by Node Manager based configurations.

Table 3-3   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	-	-	Opt	Yes
BEA_HOME	Yes	Yes	Yes	Yes
WlstScript	Yes	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
nmHome	Yes	Yes	-	-
ServerName	-	-	Yes	Yes
ServerRole	Yes	Yes	Yes	Yes
WLSUser	Yes	Yes	Yes	Yes
WLSPassword	Yes	Yes	Yes	Yes
nmUser	Yes	Yes	Yes	Yes
nmPassword	Yes	Yes	Yes	Yes
ShutdownTimeout	Opt	Opt	Opt	Opt

Table 3-3   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)
RequireAdminServer	-	-	-	Yes
AdminServerMaxWait	-	-	-	Yes
SecondLevelMonitor	0	> 0	Yes	Yes
ServerStartProgram	-	-	-	-
ServerStopProgram	-	-	-	-
User	Yes	Yes	Yes	Yes

Table 3-4 shows the attributes used by non-Node Manager based configurations.

Table 3-4   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	Opt	Opt
BEA_HOME	Yes	Yes	Yes	Yes
WlstScript	-	Yes	-	Yes
DomainDir	Yes	Yes	Yes	Yes
ListenAddressPort	Yes	Yes	Yes	Yes
MonitorProgram	Opt	Opt	Opt	Opt
nmListenAddressPort	-	-	-	-
nmType	-	-	-	-
nmHome	-	-	-	-
ServerName	Yes	Yes	Yes	Yes
ServerRole	Yes	Yes	Yes	Yes

Table 3-4　　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)
WLSUser	-	Yes	-	Yes
WLSPassword	-	Yes	-	Yes
nmUser	-	-	-	-
nmPassword	-	-	-	-
ShutdownTimeout	-	-	-	-
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

You can use sample configurations as a reference while configuring your resource.

The following list shows the types of resource configuration and the corresponding sample configuration:

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
  ```

- 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.

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.sh` and `DomainDir/bin/stopManagedWebLogic.sh`.

These stop scripts send commands to the `wlst.sh` 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 the race condition do the following:

To resolve the race issue

1   Create a copy of the *DomainDir*/bin/stopWebLogic.sh file.

2   Rename the copy as *DomainDir*/bin/stopWebLogic_old.sh.

3   In the stopWebLogic.sh file, ensure that the wlst commands are sent directly to the stdin of the wlst.sh 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.WLSTshutdown.py  2>&1
```

with the following lines:

```
echo "connect(${userID} ${password}
url='${ADMIN_URL}',adminServerName='${SERVER_NAME}')"
>"shutdown.py"
echo "shutdown('${SERVER_NAME}','Server')" >>"shutdown.py"
echo "exit()" | ${JAVA_HOME}/bin/java ${JAVA_OPTIONS}
weblogic.WLST
```

Avoiding storing unencrypted credentials in startup/shutdown scripts

Whenever you configure a WebLogic 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 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 the cluster 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 the cluster administrator starts up all the WebLogic Servers within the cluster 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:

Configuring the agent for WebLogic Server | 47
Configuring multiple Administrative Servers having the same name but different domains for Non NodeManager based
configurations

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 NodeManager based configurations

When you configure WebLogic resources with multiple administrative 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 value of "$sDomainDir". 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}
```

Enabling the agent for WebLogic Server for IMF

This chapter includes the following topics:

■ About Intelligent Monitoring Framework

■ How the agent supports intelligent resource monitoring

■ Agent functions for the IMF functionality

■ Attributes that enable IMF

■ Before you enable the agent for IMF

■ Enabling the agent for IMF

■ Disabling intelligent resource monitoring

■ Troubleshooting the configuration for IMF

■ Sample IMF configurations

■ Known issues

About Intelligent Monitoring Framework

With intelligent monitoring framework (IMF), VCS supports intelligent resource monitoring in addition to the poll-based monitoring. Poll-based monitoring polls the resources periodically whereas intelligent monitoring performs asynchronous monitoring. You can enable or disable the intelligent resource monitoring functionality of the WebLogic Server agent.

VCS process and mount-based agents use the Asynchronous Monitoring Framework (AMF) kernel driver that provides asynchronous event notifications to the agents that are enabled for Intelligent Monitoring Framework (IMF).

You can enable the WebLogic Server agent for IMF, provided the following software versions are installed:

- Veritas Cluster Server (VCS) 5.1 SP1 or later
- Veritas High Availability agent for WebLogic Server version 5.1.13.0 or later

See the *Veritas Cluster Server Administrator's Guide 5.1 Service Pack 1* for more information about:

- IMF notification module functions
- Administering the AMF kernel driver

Benefits of IMF

IMF offers the following benefits:

- Performance
  Enhances performance by reducing the monitoring of each resource at a default of 60 seconds for online resources, and 300 seconds for offline resources. IMF enables the agent to monitor a large number of resources with a minimal effect on performance.

- Faster detection
  Asynchronous notifications would detect a change in the resource state as soon as it happens. Immediate notification enables the agent to take action at the time of the event.

How the agent supports intelligent resource monitoring

When an IMF-enabled agent starts up, the agent initializes the asynchronous monitoring framework (AMF) kernel driver. After the resource is in a steady state, the agent registers with the AMF kernel driver, the details of the resource that are required to monitor the resource. For example, the agent for WebLogic Server registers the PIDs of the WebLogic Server processes with the IMF notification module. The agent's imf_getnotification function waits for any resource state changes.When the AMF kernel driver module notifies the imf_getnotification function about a resource state change, the agent framework runs the monitor agent function to ascertain the state of that resource. The agent notifies the state change to VCS, which then takes appropriate action.

See the *Veritas Cluster Server Administrator's Guide 5.1 Service Pack 1* for more information.

Agent functions for the IMF functionality

If the WebLogic Server agent is enabled for IMF, the agent supports the following functions, in addition to the functions mentioned in WebLogic Server agent functions.

imf_init

This function initializes the WebLogic Server agent to interface with the AMF kernel driver, which is the IMF notification module for the agent for WebLogic Server. This function runs when the agent starts up.

imf_getnotification

This function gets notifications about resource state changes. This function runs after the agent initializes with the AMF kernel module. This function continuously waits for notification and takes action on the resource upon notification.

imf_register

This function registers or unregisters resource entities with the AMF kernel module. This function runs for each resource after the resource goes into a steady state—online or offline.

Attributes that enable IMF

If the agent for WebLogic Server is enabled for IMF, the agent uses the following type-level attributes in addition to the attributes described in WebLogic Server agent attributes.

IMF

This resource type-level attribute determines whether the WebLogic Server agent must perform intelligent resource monitoring. You can also override the value of this attribute at the resource level.

This attribute includes the following keys:

Mode

Define this attribute to enable or disable intelligent resource monitoring. Valid values are as follows:

- 0—Does not perform intelligent resource monitoring

- 1—Performs intelligent resource monitoring for offline resources and performs poll-based monitoring for online resources

- 2—Performs intelligent resource monitoring for online resources and performs poll-based monitoring for offline resources

- 3—Performs intelligent resource monitoring for both online and for offline resources.

Note: The agent for WebLogic Server supports intelligent resource monitoring for online resources only. Hence, Mode should be set to either 0 or 2.

Type and dimension: integer-association

Default: 0

MonitorFreq

This key value specifies the frequency at which the agent invokes the monitor agent function. The value of this key is an integer.

Default: 1

You can set this key to a non-zero value for cases where the agent requires to perform both poll-based and intelligent resource monitoring.

If the value is 0, the agent does not perform poll-based process check monitoring.

After the resource registers with the AMF kernel driver, the agent calls the monitor agent function as follows:

- After every (MonitorFreq x MonitorInterval) number of seconds for online resources

- After every (MonitorFreq x OfflineMonitorInterval) number of seconds for offline resources

RegisterRetryLimit

If you enable intelligent resource monitoring, the agent invokes the imf_register agent function to register the resource with the AMF kernel driver.

The value of the RegisterRetryLimit key determines the number of times the agent must retry registration for a resource. If the agent cannot register the resource within the limit that is specified, then intelligent monitoring is disabled until the resource state changes or the value of the Mode key changes.

Default: 3.

IMFRegList

An ordered list of attributes whose values are registered with the IMF notification module.

Type and dimension: string-vector

Default: No default value

Note: The attribute values can be overriden at the resource level.

Before you enable the agent for IMF

Before you enable the WebLogic Server agent for IMF, ensure that the AMF kernel module is loaded and AMF is configured. For details, see the 'Administering the AMF kernel driver' section of the *Veritas Cluster Server Administrator's Guide 5.1 SP1*. For details about the commands you can use to configure AMF, use the `amfconfig -h` command.

Enabling the agent for IMF

In order to enable the WebLogic Server agent for IMF, you must make the following configuration changes to the attributes of the agent:

- AgentFile: Set the AgentFile attribute to Script51Agent
- IMF Mode: Set the IMF Mode attribute to 2
- IMFRegList: Update the IMFRegList attribute

The following sections provide more information on the commands you can use to make these configuration changes, depending on whether VCS is in a running state or not.

> **Note:** If you have upgraded VCS from an earlier version to version 5.1 SP1, and you already have WebLogic Server agent version 5.1.13.0 installed, ensure that you run the following commands to create appropriate symbolic links:
>
> ```
> # cd /opt/VRTSagents/ha/bin/WebLogic
> ```
> ```
> # ln -s /opt/VRTSamf/imf/imf_getnotification imf_getnotification
> ```
> ```
> # ln -s /opt/VRTSagents/ha/bin/WebLogic/monitor imf_register
> ```

If VCS is in a running state

To enable the WebLogic Server resource for IMF when VCS is in a running state:

1   Make the VCS configuration writable.

```
# haconf -makerw
```

2   Run the following command to update the AgentFile attribute.

```
# hatype -modify WebLogic AgentFile /opt/VRTSvcs/bin/Script51Agent
```

3   Run the following command to update the IMF attribute.

```
# hatype -modify WebLogic IMF Mode num MonitorFreq num
RegisterRetryLimit num
```

For example, to enable intelligent monitoring of online resources, with the MonitorFreq key set to 5, and the RegisterRetryLimit key is set to 3, run the following command:

```
# hatype -modify WebLogic IMF Mode 2 MonitorFreq 5
RegisterRetryLimit 3
```

> **Note:** The valid values for the Mode key of the IMF attribute are 0 (disabled) and 2 (online monitoring).

4   Run the following command to update the IMFRegList attribute:

```
# hatype -modify WebLogic IMFRegList BEA_HOME User ServerName
```

5   Save the VCS configuration.

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

6   If the WebLogic Server agent is running, restart the agent.

For information on the commands you can use to restart the agent, see Restarting the agent.

Restarting the agent

To restart the agent:

1   Run the following command to stop the agent forcefully:

    ```
    # haagent -stop WebLogic -force -sys <system>
    ```

Note: Stopping the agent forcefully eliminates the need to take the resource offline.

---

2   Run the following command to start the agent:

    ```
    # haagent -start WebLogic -sys <system>.
    ```

If VCS is not in a running state

To change the WebLogic type definition file when VCS is not in a running state:

1   Update the AgentFile attribute.

    ```
    static str AgentFile = "/opt/VRTSvcs/bin/Script51Agent"
    ```

2   Update the IMF attribute.

    The valid values for the Mode key of the IMF attribute are 0 (disabled) and 2 (online monitoring).

    ```
    static int IMF{} = { Mode=num, MonitorFreq=num,
    RegisterRetryLimit=num }
    ```

    For example, to update the IMF attribute such that the Mode key is set to 2, the MonitorFreq key is set to 5, and the RegisterRetryLimit key is set to 3:

    ```
    static int IMF{} = { Mode=2, MonitorFreq=5, RegisterRetryLimit=3
    }
    ```

3   Update the IMFRegList attribute.

    ```
    static str IMFRegList[] = { BEA_HOME, User, ServerName }
    ```

Disabling intelligent resource monitoring

To disable intelligent resource monitoring

1   Make the VCS configuration writable.

    ```
    # haconf -makerw
    ```

2   To disable intelligent resource monitoring for all the resources of a certain type, run the following command:

    ```
    # hatype -modify WebLogic IMF -update Mode 0
    ```

3   To disable intelligent resource monitoring for a specific resource, run the following command:

    ```
    # hares -override resource_name IMF
    ```

    ```
    # hares -modify resource_name IMF -update Mode 0
    ```

4   Save the VCS configuration.

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

Troubleshooting the configuration for IMF

If you face problems with the IMF configuration or functionality, consider the following:

- Ensure that the following attributes are configured with appropriate values.

  - AgentFile

  - IMF

  - IMFRegList
    If IMFRegList is not configured correctly, the WebLogic Server resources that have been registered for IMF get unregistered every time the monitor function is run.

- If you have configured the required attributes to enable the WebLogic Server agent for IMF, but the agent is still not IMF-enabled, restart the agent. The imf_init function runs only when the agent starts up, so when you restart the agent, imf_init runs and initializes the WebLogic Server agent to interface with the AMF kernel driver.

- You can run the following command to check the value of the MonitorMethod attribute and to verify that a resource is registered for IMF.

    ```
    # hares -value resource MonitorMethod system
    ```

The MonitorMethod attribute specifies the monitoring method that the agent uses to monitor the resource:

- Traditional—Poll-based resource monitoring

- IMF—Intelligent resource monitoring

■ You can use the `amfstat` to see a list of registered PIDs for a WebLogic resource. The output of the `ps -ef` command for the AdminServer process.

```
$ /usr/ucb/ps auxww | egrep 'AdminServer|MS1'

root    7241  0.1  2.8421168284184 ?   S 17:44:47  2:22 /opt/
jdk1.6.0_25/bin/java -client -Xms256m -Xmx512m -XX:Compile
Threshold=8000 -XX:PermSize=48m -XX:MaxPermSize=128m -
Dweblogic.Name=AdminServer -Djava.security.policy=/var/Oracle/
Middleware/wlserver_10.3/server/lib/weblogic.policy -Xverify:
none -da -Dplatform.home=/var/Oracle/Middleware/wlserver_10.3
 -Dwls.home=/var/Oracle/Middleware/wlserver_10.3/server -
Dweblogic.home=/var/Oracle/Middleware/wlserver_10.3/server
-Dweblogic.management.discover=true -Dwlw.iterativeDev= -
Dwlw.testConsole= -Dwlw.logErrorsToConsole= -Dweblogic.ext.
dirs=/var/Oracle/Middleware/patch_wls1034/profiles/
default/sysext_manifest_classpath:/var/Oracle/Middleware/
patch_ocp360/profiles/default/sysext_manifest_classpath
weblogic.Server

root  13869  0.1 2.2389816223008 pts/1  S 14:47:43  0:23 /opt/
jdk1.6.0_25/bin/java -client -Xms256m -Xmx512m -XX:Compile
Threshold=8000 -XX:PermSize=48m -XX:MaxPermSize=128m -
Dweblogic.Name=MS1 -Djava.security.policy=/var/Oracle/
Middleware/wlserver_10.3/server/lib/weblogic.policy -
Dweblogic.security.SSL.trustedCAKeyStore=/var/Oracle/
Middleware/wlserver_10.3/server/lib/cacerts -
Xverify:none -da -Dplatform.home=/var/Oracle/Middleware/
wlserver_10.3 -Dwls.home=/var/Oracle/Middleware/wlserver_10.3/
server -Dweblogic.home=/var/Oracle/Middleware/wlserver_10.3/
server -Dweblogic.management.discover=false -Dweblogic.
management.server=http://10.209.73.87:8001 -Dwlw.iterativeDev
=false -Dwlw.testConsole=false -Dwlw.logErrorsToConsole=false
 -Dweblogic.ext.dirs=/var/Oracle/Middleware/patch_wls1034/
profiles/default/sysext_manifest_classpath:/var/Oracle/
Middleware/patch_ocp360/profiles/default/sysext_manifest
_classpath weblogic.Server
```

The `amfstat` command shows the PIDs monitored by the WebLogic Server agent.

```
$ /opt/VRTSamf/bin/amfstat

AMF Status Report

Registered Reapers (1):
======================
 RID    PID     MONITOR    TRIGG    REAPER
 2      6505    2          0        WebLogic

Process ONLINE Monitors (2):
============================
 RID       R_RID    PID        GROUP
 9         2        7241       wls_admin
 11        2        13869      wls_managed
```

■ Run the following command to set the ResLogLevel attribute to TRACE. When you set ResLogLevel to TRACE, the agent logs messages in the WebLogic_A.log file.

```
# hares -modify Queue1 ResLogLevel TRACE
```

■ Run the following command to view the content of the AMF in-memory trace buffer.

```
# amfconfig -p dbglog
```

For more troubleshooting information, see the following:

■ See "*Troubleshooting the agent for WebLogic Server*" on page 67.

■ For information about asynchronous monitoring, attributes that enable asynchronous monitoring, and instructions on using some AMF CLI: *Veritas Cluster Server Administrator's Guide 5.1 Service Pack 1*

■ *Veritas Cluster Server Release Notes 5.1 Service Pack 1*

Sample IMF configurations

An example of a type definition file for a WebLogic Server agent that is IMF-enabled is as follows.

In this example, the IMF-related attributes are set to the following values:

AgentFile   /opt/VRTSvcs/bin/Script51Agent

IMF{}	{ Mode=2, MonitorFreq=5, RegisterRetryLimit=3 }
IMFRegList[]	{ BEA_HOME, User, ServerName }
LevelTwoMonitorFreq	25

```
type WebLogic (
     static str AgentDirectory = "/opt/VRTSagents/ha/bin/WebLogic"
     static str AgentFile = "/opt/VRTSvcs/bin/Script50Agent"
     static int LevelTwoMonitorFreq = 25
     static int RestartLimit = 3
     static str ArgList[] = { ResLogLevel, State, IState, AdminURL,
     BEA_HOME, WlstScript, DomainDir, ListenAddressPort,
     MonitorProgram, nmListenAddressPort, nmType, nmHome, ServerName,
     ServerRole, User, WLSUser, WLSPassword, nmUser, nmPassword,
     ServerStartProgram, ServerStopProgram, ShutdownTimeout,
     RequireAdminServer, AdminServerMaxWait, SecondLevelMonitor }
     static boolean AEPTimeout = 1
     static int IMF{} = { Mode=2, MonitorFreq=5, RegisterRetryLimit=3 }
     static str IMFRegList[] = { BEA_HOME, User, ServerName }
     str ResLogLevel = INFO
     str AdminURL
     str BEA_HOME
     str WlstScript
     str DomainDir
     str ListenAddressPort
     str MonitorProgram
     str nmListenAddressPort
     str nmType = ssl
     str nmHome
     str ServerName
     str ServerRole
     str User
     str WLSUser
     str WLSPassword
     str nmUser
     str nmPassword
     int ShutdownTimeout
     str ServerStartProgram
     str ServerStopProgram
     boolean RequireAdminServer = 0
     int AdminServerMaxWait = 60
     int SecondLevelMonitor
```

```
)
```

A sample resource configuration from the `/etc/VRTSvcs/conf/config/main.cf` file is as follows:

```
WebLogic wls_admin (
    Critical = 0
    BEA_HOME = "/var/Oracle/Middleware"
    WlstScript = "/var/Oracle/Middleware/wlserver_10.3/common/bin/
    wlst.sh"
    DomainDir = "/var/Oracle/Middleware/user_projects/domains/domain1"
    ListenAddressPort = "10.209.73.87:8001"
    ServerName = AdminServer
    ServerRole = Administrative
    User = root
    ServerStartProgram = "/var/Oracle/Middleware/user_projects/domains/
    domain1/bin/startWebLogic.sh"
    ServerStopProgram = "/var/Oracle/Middleware/user_projects/domains/
    domain1/bin/stopWebLogic.sh"
    )
```

Known issues

This release of the agent for WebLogic Server has the following known issue:

Problem

An error message might appear when you run the `hares -offline` command to take a resource offline.

Description

When a resource is taken offline, it is unregistered from the AMF module. However, the imf_register function attempts to unregister the resource again. This results in the following error message from the engine log.

```
VCS ERROR V-16-2-13710 Thread(1029) Resource(WLSAdmin_res) -
imf_register entry point failed with exit code(1)
```

The following message is logged in the agent log:

```
VCS INFO V-16-55000-10199 Resource(WLSAdmin_res) -
(lopban26:imf_register) Sys:RunWithEnvCmdWithOutputWithTimeOut:Going
to run command line [/opt/VRTSamf/bin/amfregister -u -rWebLogic -g
WLSAdmin_res ], as User [root] VCS INFO V-16-55000-10209
Resource(WLSAdmin_res) - (lopban26:imf_register)
```

```
Sys:RunWithEnvCmdWithOutputWithTimeOut:Command line
[/opt/VRTSamf/bin/amfregister -u -rWebLogic -g WLSAdmin_res ] provided
a non-zero exit code -- this does not necessarily indicate a
problem... (Perl's OS error variable prior to the command-pipe close
was [], and after the close was [] ) VCS INFO V-16-55000-10289
Resource(WLSAdmin_res) - (lopban26:imf_register)
VCSagentFW:messageEngineLog:[AMF libvxamf NOTICE Ignoring the group
unregister request; group named \"WLSAdmin_res\" not found]
```

Workaround

It is safe to ignore this error message.

Configuring the service groups for WebLogic Server

This chapter includes the following topics:

- Before configuring the service groups for WebLogic Server
- Configuring service groups for WebLogic Server

Before configuring the service groups for WebLogic Server

Before you configure the WebLogic Server service group, you must:

- Verify that VCS is installed and configured on all nodes in the cluster where you will configure the service group.
  Refer to the *Veritas Cluster Server Installation Guide* for more information.

- Verify that the Veritas agent for WebLogic Server is installed on all nodes in the cluster.
  See "Installing the agent in a VCS environment" on page 22.

Configuring service groups for WebLogic Server

Assuming that the target implementation has licensed the Veritas Storage Foundation and High Availability products, perform the following steps to cluster an instance of WebLogic Server:

To configure the service groups for WebLogic Server

1   Create UNIX user and group accounts.

Create a UNIX username in the cluster namespace (NIS, NIS+, LDAP or the local password files) for WebLogic Server operations. Ensure that all cluster nodes use the same user with the same user UID and default shell.

Symantec recommends the use of the local configuration files over naming services like NIS, NIS+ or LDAP for the reason that name resolution using a centralized service takes additional time and is subject to network delays. If the local file approach is used, ensure that all nodes are updated with the exact same information to guarantee consistency throughout the cluster. Also make sure the name service resolution configuration (/etc/nsswitch.conf on most UNIX systems) gives preference to the local files over centralized naming services.

2   Create the supporting directory structure.

A well-designed directory structure for your WebLogic Server instances simplifies the cluster configuration and creates a storage environment that is intuitive and easier to manage. Assuming that all WebLogic Server instances will be clustered and installed on shared disk, Symantec recommends a directory structure similar to the following:

Directory	Purpose
/wls90	Root directory in which to group all WebLogic Server instances supporting a particular domain.
/wls90/admin	Path used to mount the file system dedicated for the WebLogic Administration Server program and configuration files. All WebLogic binaries and configuration files for this Administration Server are stored in this file system.
/wls90/mng01	Path used to mount the file system dedicated for WebLogic Managed Server 1 program and configuration files. All WebLogic binaries and configuration files for Managed Server 1 are stored in this file system.
/wls90/mng02	Path used to mount the file system dedicated for WebLogic Managed Server 2 program and configuration files. All WebLogic binaries and configuration files for Managed Server 2 are stored in this file system.

Additional notes about the sample directory structure

- This sample directory structure is for WebLogic Server 9. It includes directories for only two WebLogic Managed Servers, but the naming structure supports an unlimited number.

- The directories and subdirectories are created on the root file system on each system in the cluster. The mount points need to exist on all systems in the cluster that are configured to run the WebLogic Server instance.

- The sub-directories under /wls90 are mount points on which file systems will be mounted. These file systems are stored on shared disks. Each WebLogic Server instance is installed on its own dedicated file system; it is not installed in the root file system.

3   Create high level mount points for WebLogic Server operations.

4   Create a disk group and volume.

Consult the *Veritas Volume Manager Administrator's Guide* for details on how to provision disk group and volume resources.

5   Create the file system.

6   Create a Virtual IP Address.

Provision a Virtual IP address in the network namespace (i.e. NIS, NIS+ or LDAP). Ensure the IP address and host name pair are defined for all nodes in the cluster. If the IP and host name pair are defined in the local host map, make sure all cluster nodes have the same host map record.

7   Create service group and resources on a cluster.

Create a service group on a cluster and define resources for the NIC, IP, DiskGroup, and Mount resources. Consult the cluster documentation for detailed information on NIC, IP, DiskGroup, and Mount resource types.

Online these newly created resources on one node in the cluster.

8   Install and configure WebLogic Server.

Install the WebLogic software on the newly created and mounted file system. After it is installed, change the file and group ownership to reflect the WebLogic Server UNIX user and group accounts created earlier.

Modify the WebLogic Server configuration to use the Virtual IP address and port. Refer the BEA WebLogic Server documentation for instructions to bind a WebLogic Server instance to its dedicated virtual IP address and port number. Configuring the WebLogic Server to bind is essential to ensure that it always listens on the same virtual IP address and port number regardless of the system in the cluster on which it is running.

9   Finalize and test the configuration as follows:

■ Create the WebLogic Server resource.

■ Online the newly created resource.

■ Test instance startup, shutdown and switchover as required, confirming overall availability requirements.

To refer to a sample configuration Service Group:

See "Sample service group configuration for WebLogic Server" on page 80.

Troubleshooting the agent for WebLogic Server

This chapter includes the following topics:

- Using the correct software and operating system versions
- Meeting prerequisites
- Configuring WebLogic Server resources
- Starting the WebLogic Server instance outside a cluster
- Defining additional environment variables before starting or stopping WebLogic resources
- Reviewing error log files
- Problems starting a Managed Server through the administrative console
- Unable to bring two or more VCS resources offline simultaneously
- Serial version UID mismatch on the AIX platform

Using the correct software and operating system versions

Ensure that no issues arise due to incorrect software and operating system versions. For the correct versions of operating system and software to be installed on the resource systems:

See "Supported software" on page 12.

Meeting prerequisites

Before installing the agent for WebLogic Server, double check that you meet the prerequisites.

For example, you must install the ACC library on VCS before installing the agent for WebLogic Server.

See "Before you install the Veritas agent for WebLogic Server" on page 19.

Configuring WebLogic Server resources

Before using WebLogic Server resources, ensure that you configure the resources properly. For a list of attributes used to configure all WebLogic Server resources, refer to the agent attributes.

Starting the WebLogic Server instance outside a cluster

If you face problems while working with a resource, you must disable the resource within the cluster framework. A disabled resource is not under the control of the cluster framework, and so you can test the WebLogic Server instance independent of the cluster framework. Refer to the cluster documentation for information about disabling a resource.

You can then restart the WebLogic Server instance outside the cluster framework.

Note: Use the same parameters that the resource attributes define within the cluster framework while restarting the resource outside the cluster framework.

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

To restart a Node Manager outside the cluster framework

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
WlstScript	/Oracle/Middleware/wlserver_10.3/common/bin/wlst.sh
nmListenAddressPort	wls90admsol:5556
nmType	ssl
nmHome	/Oracle/Middleware/wlserver_10.3/common/nodemanager
ServerRole	NodeManager

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 cluster framework, you can then attempt to implement the Node Manager within the cluster framework.

To restart a Managed or Administrative Server outside the cluster framework

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
WlstScript	/Oracle/Middleware/wlserver_10.3/common/bin/wlst.sh
nmListenAddressPort	wls90admsol:5556
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 cluster framework, you can then attempt to implement the server within the cluster framework.

Defining additional environment variables before starting or stopping WebLogic resources

By default, WebLogic uses the `commEnv.sh` file, which contains the required environment that WebLogic Server needs.

This file is sourced in the WebLogic `start` and `stop` scripts. For example:

```
bash-3.00# grep -i env startWebLogic.sh

# environment by calling @USERDOMAINHOME/setDomainEnv.

# setDomainEnv initializes or calls commEnv to initialize the
following variables:
```

■ `# Call setDomainEnv here.`

■ `. ${DOMAIN_HOME}/bin/setDomainEnv.sh $*`

As a standard followed by all application agents, the WebLogic Server agent does not source the user's profile. If you need to export any additional environment

variables, create an environment file with the required variables and source it in the `start` and `stop` scripts. For example:

After you source the `setDomainEnv.sh` environment file, append a line in `startWebLogic.sh` in the following manner in the start/stop scripts:

```
# Call User Environment here.

. <Path to env file>/setUserEnv.sh
```

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 cluster log files

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

■ The VCS engine log file is /var/VRTSvcs/log/engine_A.log.

Using trace level logging

The ResLogLevel attribute controls the level of logging that is written in a cluster 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:

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

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

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

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:

```
# 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.

You can also contact Symantec support for more help.

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

If a resource, WLS90Mng01_nodemanager is unable to bring a WebLogic Node Manager online, you can access the /tmp/.VRTSWebLogic/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
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://download.oracle.com/docs/cd/E13222_01/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.

See "Editing the WebLogic stop script" on page 44.

Serial version UID mismatch on the AIX platform

BEA Systems have identified a serial version UID mismatch issue while using a
WebLogic Server version 9.1 on the AIX platform. For information about the issue:

http://download.oracle.com/docs/cd/E13196_01/platform/suppconfigs/
configs/ibm_aix/ibm_aix53.html#1061399

You can fix the issue for the WebLogic Servers that the Node Manager starts.

To fix the issue for an administrative server

1   Go to the *DomainDir*/servers/*AdminServerName*/data/nodemanager directory.

2   Create a startup.properties file.

3   Add this line to the startup.properties file:

```
Arguments = -
Dcom.sun.xml.namespace.QName.useCompatibleSerialVersionUID=1.0
```

4   Save the startup.properties file.

To fix the issue for a managed server

1 Access the Administrative Server console.

2 Go to the Server Start settings.

3 In the Arguments field, add this line:

```
-Dcom.sun.xml.namespace.QName.useCompatibleSerialVersionUID=1.0
```

Sample Configurations

This appendix includes the following topics:

- About sample configurations for the agent for WebLogic Server

- Configuring "weblogic.Admin GETSTATE" based monitoring

- Sample agent type definition for WebLogic server

- Sample service group configuration for WebLogic Server

- Sample resource configurations for WebLogic Server

- Service group dependencies for WebLogic Server

- Sample configuration in a VCS environment

About sample configurations for the agent for WebLogic Server

The sample configuration graphically depicts the resource types, resources, and resource dependencies within the service group. Review these dependencies carefully before configuring the agent for WebLogic Server. For more information about these resource types, see the *Veritas Cluster Server Bundled Agents Reference Guide.*

Configuring "weblogic.Admin GETSTATE" based monitoring

Configure the WebLogic agent to perform Second Level Monitoring using the weblogic.Admin command to obtain the state of the WebLogic Server. Note that

the examples and process assumes the WebLogic user account's default shell is /bin/sh.

To configure "weblogic.Admin GETSTATE" based monitoring

1   Log into a system that has the desired WebLogic file systems mounted. Make sure you login as the WebLogic User and make the ScriptDir directory the current working directory.

```
# cd DomainDir
```

2   Read in the environment file specified by the EnvFile attribute:

```
# . EnvFile
```

3   Create the required WebLogic authentication credential files using the following WebLogic command format:

```
# java weblogic.Admin\
-username weblogicUser\
-password weblogicUserPassword
-userconfigfile ./VRTSWebLogicConfig.properties\
-userkeyfile ./VRTSWebLogicKey.properties STOREUSERCONFIG
```

Example: Assuming the WebLogic user is 'weblogic' with a password 'weblogic', you would expect to see the following:

```
# java weblogic.Admin\
-username weblogic \
-password weblogic \ -userconfigfile
./VRTSWebLogicConfig.properties \ -userkeyfile
./VRTSWebLogicKey.properties STOREUSERCONFIG
```

The following message is displayed:

```
Creating the key file can reduce the security of your system if
it is not kept in a secured location after it is created. Do you
want to create the key file? y or n
```

4   Press **y**.

5   Use the weblogic.Admin command to test the GETSTATE option using the newly created authentication credential property files. The GETSTATE command format is listed as follows:

```
java weblogic.Admin -url t3://<Host>:<Port>\
-userconfigfile ./VRTSWebLogicConfig.properties\
-userkeyfile ./VRTSWebLogicKey.properties GETSTATE
```

Example: Assuming the WebLogic server was online configured to use the Virtual IP address 10.136.228.77 with port 7001, you would expect to see the following:

```
java weblogic.Admin -url t3://10.136.228.77:7001\
-userconfigfile ./VRTSWebLogicConfig.properties\
-userkeyfile ./VRTSWebLogicKey.properties GETSTATE
Current state of "AdminServer" : RUNNING
```

Sample agent type definition for WebLogic server

Examples of agent type definition files are:

```
type WebLogic (
static str AgentDirectory = "/opt/VRTSagents/ha/bin/WebLogic"
static str AgentFile = "/opt/VRTSvcs/bin/Script50Agent"
static int RestartLimit = 3
static str ArgList[] = { ResLogLevel, State, IState, AdminURL,
BEA_HOME, WlstScript, DomainDir, ListenAddressPort, MonitorProgram,
nmListenAddressPort, nmType, nmHome, ServerName, ServerRole, User,
WLSUser, WLSPassword, nmUser, nmPassword, ServerStartProgram,
ServerStopProgram, ShutdownTimeout , RequireAdminServer,
AdminServerMaxWait, SecondLevelMonitor }
static boolean AEPTimeout = 1
str ResLogLevel = INFO
str AdminURL
str BEA_HOME
str WlstScript
str DomainDir
str ListenAddressPort
str MonitorProgram
str nmListenAddressPort
str nmType = ssl
str nmHome
```

```
str ServerName
str ServerRole
str User
str WLSUser
str WLSPassword
str nmUser
str nmPassword
int ShutdownTimeout
str ServerStartProgram
str ServerStopProgram
boolean RequireAdminServer = 0
int AdminServerMaxWait = 60
int SecondLevelMonitor
)
```

Sample service group configuration for WebLogic Server

A WebLogic Server resource consists of the following:

Disk Group: Veritas Volume Manager disk group contains information required by the DiskGroup agent to import and export the shared disk object used in support of a clustered WebLogic Server instance. While the use of shared disk is not required to cluster an instance of WebLogic Server, Symantec recommends the use of a shared volume to eliminate the requirement to synchronize local copies of the WebLogic Server binaries and configuration files on each node in a multi-node cluster.

Mount: This resource mounts, monitors, and unmounts the file system that is dedicated to the WebLogic Server installation and configuration files. Use the resource type Mount to create this resource.

Network Interface: This resource monitors the network interface card through which the WebLogic Server communicates with other services.

Virtual IP: This resource configures the virtual IP address dedicated to the WebLogic Server. External services, programs, and clients use this address to communicate with this WebLogic Server instance.

WebLogic Server: This resource starts, stops, and monitors the WebLogic Server instance. Use the WebLogic Server resource type to create this resource.

Figure A-1 shows an example of a single service group with an Administrative Server.

Figure A-1   Service group configuration with Administrative server

WLS90Admin_admin

Server

WLS90Admin_nodemanager

Server

WLS90Admin_IP

IP

WLS90Admin_mount   *Mount*

DiskGroup

WLS90Admin_dg
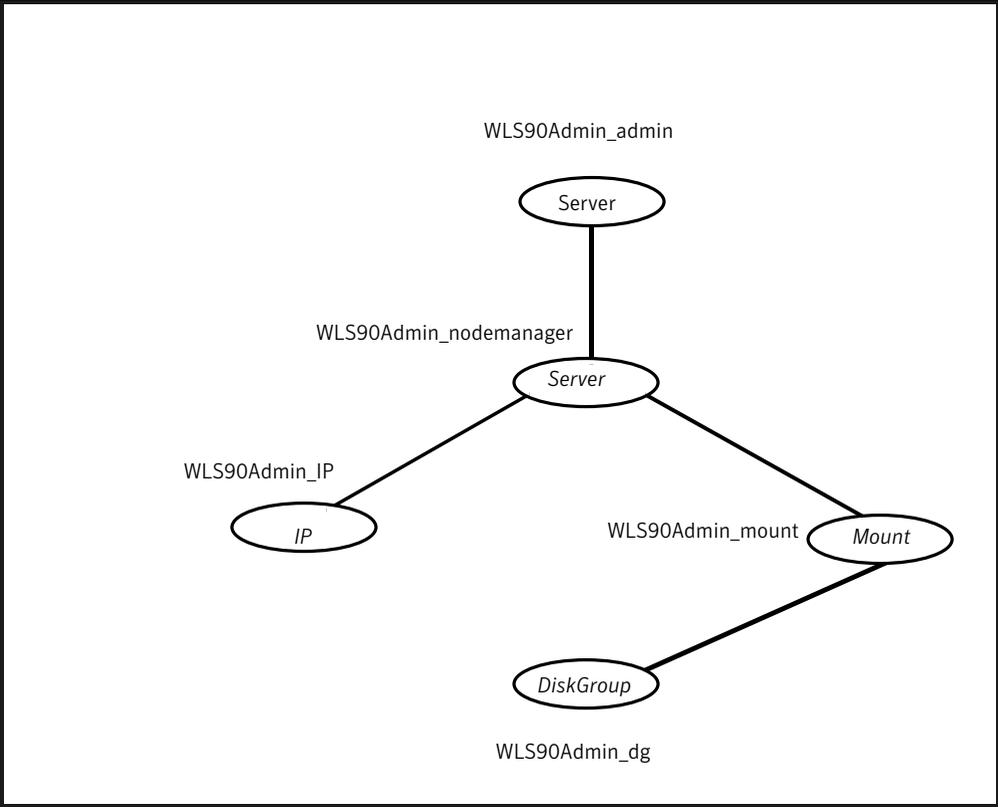
Figure A-2 shows a service group with Administrative and Managed Servers.

Figure A-2   Service group configuration with Administrative and Managed servers
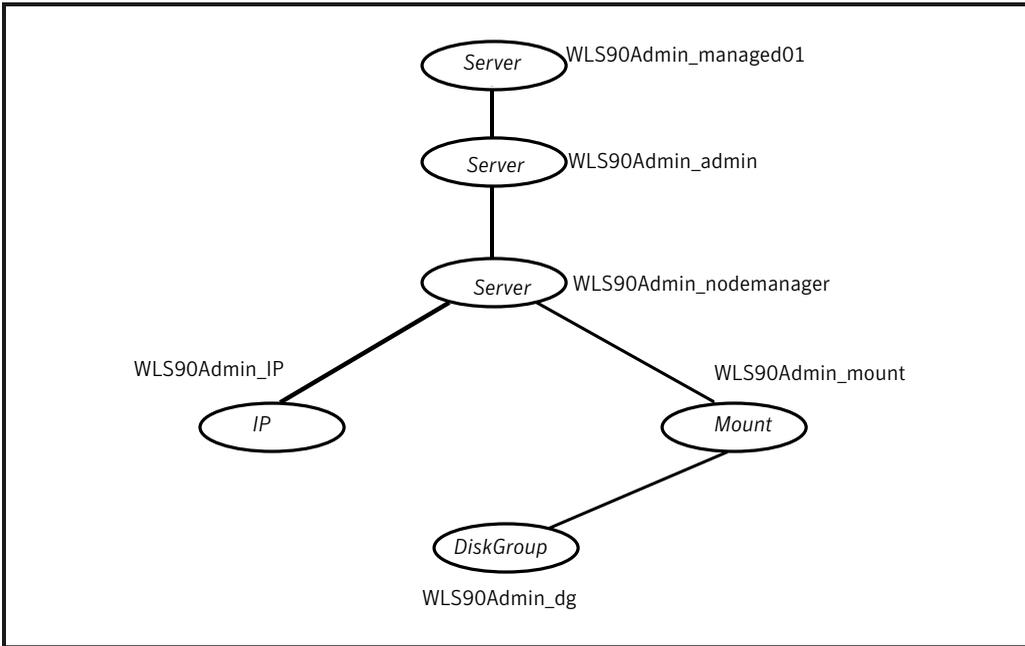
Sample resource configurations for WebLogic Server

The sample resource configurations for WebLogic Server are shown in the following sections.

Node Manager without SLM enabled

Table A-1 depicts a typical configuration for Node Manager with second level monitoring (SLM) not enabled.

Table A-1   Node Manager without SLM enabled

Attribute	Value
ResLogLevel	INFO
AdminURL	
BEA_HOME	/Oracle/Middleware

Table A-1   Node Manager without SLM enabled *(continued)*

Attribute	Value
WlstScript	/Oracle/Middleware/wlserver_10.3/common/bin/wlst.sh
ListenAddressPort	
MonitorProgram	
nmListenAddressPort	wlsadmin1:5556
nmType	ssl
nmHome	/Oracle/Middleware/wlserver_10.3/common/nodemanager
ServerName	
ServerRole	NodeManager
WLSUser	weblogic
WLSPassword	EQFsHqkkMNRkL
nmUser	weblogic
nmPassword	EQFsHqkkMNRkL
ShutdownTimeout	0
ServerStartProgram	
ServerStopProgram	
RequireAdminServer	false
AdminServerMaxWait	60
SecondLevelMonitor	0

Node Manager with SLM enabled

Table A-2 depicts a typical configuration for Node Manager with second level monitoring (SLM) enabled.

Table A-2   Node Manager with SLM enabled

Attribute	Value
ResLogLevel	INFO

Table A-2   Node Manager with SLM enabled *(continued)*

Attribute	Value
AdminURL	
BEA_HOME	/Oracle/Middleware
WlstScript	/Oracle/Middleware/wlserver_10.3/common/bin/wlst.sh
DomainDir	/Oracle/Middleware/user_projects/domains/wls_domain1
ListenAddressPort	
MonitorProgram	
nmListenAddressPort	wlsadmin1:5556
nmType	ssl
nmHome	/Oracle/Middleware/wlserver_10.3/common/nodemanager
ServerName	
ServerRole	NodeManager
WLSUser	weblogic
WLSPassword	EQFsHqkkMNRkL
nmUser	weblogic
nmPassword	EQFsHqkkMNRkL
ShutdownTimeout	0
ServerStartProgram	
ServerStopProgram	
RequireAdminServer	false
AdminServerMaxWait	60
SecondLevelMonitor	1

Administrative Server (NM) without SLM enabled

Table A-3 depicts a typical configuration for Administrative server (NM) with second level monitoring (SLM) not enabled.

Table A-3   Administrative Server (NM) without SLM enabled

Attribute	Value
ResLogLevel	INFO
AdminURL	
BEA_HOME	/Oracle/Middleware
WlstScript	/Oracle/Middleware/wlserver_10.3/common/bin/wlst.sh
DomainDir	/Oracle/Middleware/user_projects/domains/wls_domain
ListenAddressPort	wlsadmin:7011
MonitorProgram	
nmListenAddressPort	wlsadmin:5556
nmType	ssl
nmHome	/Oracle/Middleware/wlserver_10.3/common/nodemanager
ServerName	AdminServer
ServerRole	Administrative
WLSUser	weblogic
WLSPassword	EQFsHqkkMNRkL
nmUser	weblogic
nmPassword	EQFsHqkkMNRkL
ShutdownTimeout	0
ServerStartProgram	
ServerStopProgram	
RequireAdminServer	false
AdminServerMaxWait	60
SecondLevelMonitor	0

Administrative Server (NM) with SLM enabled

Table A-4 depicts a typical configuration for Administrative Server (NM) with the second level monitoring (SLM) enabled.

Table A-4   Administrative Server (NM) with SLM enabled

Attribute	Value
ResLogLevel	INFO
AdminURL	
BEA_HOME	/Oracle/Middleware
WlstScript	/Oracle/Middleware/wlserver_10.3/common/bin/wlst.sh
DomainDir	/Oracle/Middleware/user_projects/domains/wls_domain
ListenAddressPort	wlsadmin:7011
MonitorProgram	
nmListenAddressPort	wlsadmin:5556
nmType	ssl
nmHome	/Oracle/Middleware/wlserver_10.3/common/nodemanager
ServerName	AdminServer
ServerRole	Administrative
WLSUser	weblogic
WLSPassword	EQFsHqkkMNRkL
nmUser	weblogic
nmPassword	EQFsHqkkMNRkL
ShutdownTimeout	0
ServerStartProgram	
ServerStopProgram	
RequireAdminServer	false
AdminServerMaxWait	60
SecondLevelMonitor	3

Managed Server (NM) without SLM enabled

Table A-5 depicts a typical configuration for Managed Server (NM) with second level monitoring (SLM) not enabled.

Table A-5   Managed Server (NM) without SLM enabled

Attribute	Value
ResLogLevel	INFO
AdminURL	http://wlsadmin:7011
BEA_HOME	/Oracle/Middleware
WlstScript	/Oracle/Middleware/wlserver_10.3/common/bin/wlst.sh
DomainDir	/Oracle/Middleware/user_projects/domains/wls_domain
ListenAddressPort	wlsadmin:7012
MonitorProgram	
nmListenAddressPort	wlsadmin:5556
nmType	ssl
nmHome	/Oracle/Middleware/wlserver_10.3/common/nodemanager
ServerName	ManagedServer01
ServerRole	Managed
WLSUser	weblogic
WLSPassword	EQFsHqkkMNRkL
nmUser	weblogic
nmPassword	EQFsHqkkMNRkL
ShutdownTimeout	0
ServerStartProgram	
ServerStopProgram	
RequireAdminServer	false
AdminServerMaxWait	15
SecondLevelMonitor	0

Managed Server (NM) with SLM enabled

Table A-6 depicts a typical configuration for Managed Server (NM) with second level monitoring (SLM) enabled.

Table A-6   Managed Server (NM) with SLM enabled

Attribute	Value
ResLogLevel	INFO
AdminURL	http://wlsadmin:7011
BEA_HOME	/Oracle/Middleware
WlstScript	/Oracle/Middleware/wlserver_10.3/common/bin/wlst.sh
DomainDir	/Oracle/Middleware/user_projects/domains/wls_domain
ListenAddressPort	wlsadmin:7012
MonitorProgram	
nmListenAddressPort	wlsadmin:5556
nmType	ssl
nmHome	/Oracle/Middleware/wlserver_10.3/common/nodemanager
ServerName	ManagedServer01
ServerRole	Managed
WLSUser	weblogic
WLSPassword	EQFsHqkkMNRkL
nmUser	weblogic
nmPassword	EQFsHqkkMNRkL
ShutdownTimeout	0
ServerStartProgram	
ServerStopProgram	
RequireAdminServer	false
AdminServerMaxWait	15
SecondLevelMonitor	1

Managed Server (NNM) without SLM enabled

Table A-7 depicts a typical configuration for Managed Server (NNM) with the second level monitoring (SLM) not enabled.

Table A-7 Managed Server (NNM) without SLM enabled

Attribute	Value
ResLogLevel	INFO
AdminURL	t3://wlsadmin:7001
BEA_HOME	/Oracle/Middleware
WlstScript	/Oracle/Middleware/wlserver_10.3/common/bin/wlst.sh
DomainDir	
ListenAddressPort	
MonitorProgram	
nmListenAddressPort	
nmType	ssl
nmHome	
ServerName	ManagedServer01
ServerRole	Managed
WLSUser	
WLSPassword	
nmUser	
nmPassword	
ShutdownTimeout	0
ServerStartProgram	/Oracle/Middleware/user_projects/domains/wls_domain/bin/ startManagedWebLogic.sh
ServerStopProgram	/Oracle/Middleware/user_projects/domains/wls_domain/bin/ stopManagedWebLogic.sh
RequireAdminServer	false
AdminServerMaxWait	15
SecondLevelMonitor	0

Managed Server (NNM) with SLM enabled

Table A-8 depicts a typical configuration for Managed server (NNM) with second level monitoring (SLM) enabled.

Table A-8   Managed Server (NNM) with SLM enabled

Attribute	Value
ResLogLevel	INFO
AdminURL	t3://wlsadmin:7001
BEA_HOME	/Oracle/Middleware
WlstScript	/Oracle/Middleware/wlserver_10.3/common/bin/wlst.sh
DomainDir	
ListenAddressPort	wlsadmin:7011
MonitorProgram	
nmListenAddressPort	
nmType	ssl
nmHome	
ServerName	ManagedServer01
ServerRole	Managed
WLSUser	weblogic
WLSPassword	EQFsHqkkMNRkL
nmUser	weblogic
nmPassword	EQFsHqkkMNRkL
ShutdownTimeout	0
ServerStartProgram	/Oracle/Middleware/user_projects/domains/wls_domain/bin/ startManagedWebLogic.sh
ServerStopProgram	/Oracle/Middleware/user_projects/domains/wls_domain/bin/ stopManagedWebLogic.sh
RequireAdminServer	false

Table A-8   Managed Server (NNM) with SLM enabled *(continued)*

Attribute	Value
AdminServerMaxWait	60
SecondLevelMonitor	1

Administrative Server (NNM) without SLM enabled

Table A-9 depicts a typical configuration for Administrative server (NNM) with second level monitoring (SLM) not enabled.

Table A-9   Administrative Server (NNM) without SLM enabled

Attribute	Value
ResLogLevel	INFO
AdminURL	
BEA_HOME	/Oracle/Middleware
WlstScript	/Oracle/Middleware/wlserver_10.3/common/bin/wlst.sh
DomainDir	
ListenAddressPort	wls90admsol:7011
MonitorProgram	
nmListenAddressPort	
nmType	ssl
nmHome	
ServerName	AdminServer
ServerRole	Administrative
WLSUser	
WLSPassword	
nmUser	
nmPassword	
ShutdownTimeout	0

| Table A-9 | Administrative Server (NNM) without SLM enabled *(continued)* |

Attribute	Value
ServerStartProgram	/Oracle/Middleware/user_projects/domains/wls_domain/bin/ startWebLogic.sh
ServerStopProgram	/Oracle/Middleware/user_projects/domains/wls_domain/bin/ stopWebLogic.sh
RequireAdminServer	false
AdminServerMaxWait	60
SecondLevelMonitor	0

Administrative Server (NNM) with SLM enabled

Table A-10 depicts a typical configuration for Administrative Server (NNM) with the second level monitoring (SLM) enabled.

| Table A-10 | Administrative Server (NNM) with SLM enabled |

Attribute	Value
ResLogLevel	INFO
AdminURL	
BEA_HOME	/Oracle/Middleware
WlstScript	/Oracle/Middleware/wlserver_10.3/common/bin/wlst.sh
DomainDir	
ListenAddressPort	wlsadmin:7001
MonitorProgram	
nmListenAddressPort	
nmType	ssl
nmHome	
ServerName	AdminServer
ServerRole	Administrative
WLSUser	weblogic

Table A-10   Administrative Server (NNM) with SLM enabled *(continued)*

Attribute	Value
WLSPassword	EQFsHqkkMNRkL
nmUser	
nmPassword	
ShutdownTimeout	0
ServerStartProgram	/Oracle/Middleware/user_projects/domains/wls_domain/bin/ startWebLogic.sh
ServerStopProgram	/Oracle/Middleware/user_projects/domains/wls_domain/bin/ stopWebLogic.sh
RequireAdminServer	false
AdminServerMaxWait	15
SecondLevelMonitor	1

Service group dependencies for WebLogic Server

Cluster administrators use Service Group dependencies to create links between unrelated Service Group objects within a cluster. In this version of WebLogic Server, you no longer require Service Group dependencies.

The Managed Server online operation can automatically perform an Administrative Server probe. So even though Managed Server instances depend on the domain Administrative Server instance, you can have a Service Group with Managed Servers only.

See "Delaying managed server startup process" on page 46.

Figure A-3 shows a single Service Group looks with Managed Servers only.

Figure A-3   Single Service group with Managed Servers only
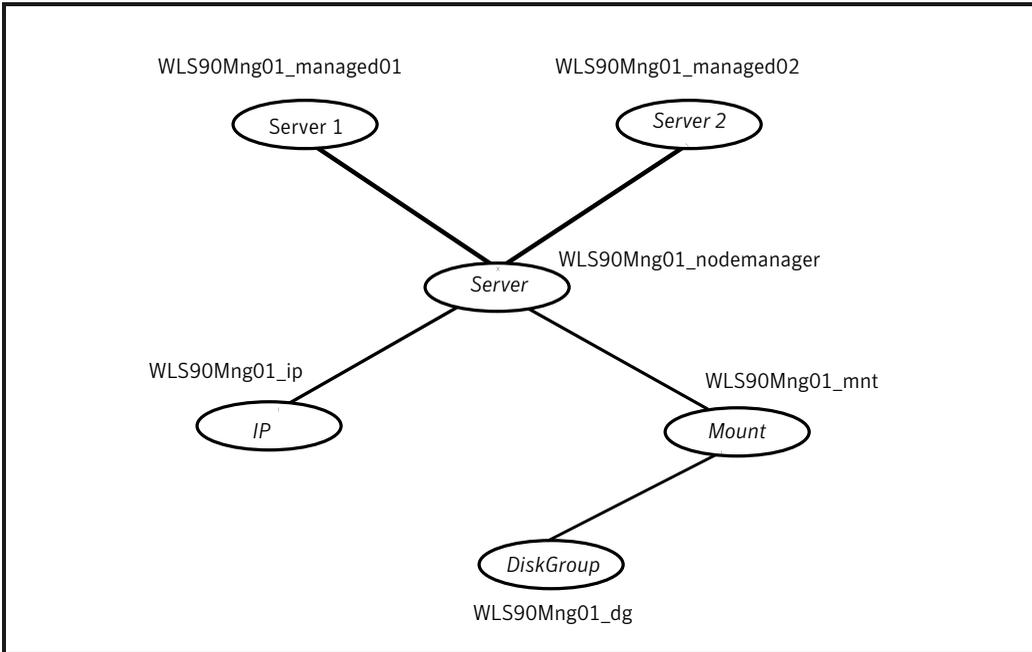
Sample configuration in a VCS environment

To provide a complete example, the following main.cf excerpt from a Solaris cluster defines a Service Group to support one WebLogic Server instance.

```
group wlsadmin
(
SystemList = { systemA = 1, systemB = 2 }
)

DiskGroup wlsadmin_dg
(
    DiskGroup = wlsadmin
)

Mount wlsadminmnt
(
    MountPoint = "/wls90/admin"
    BlockDevice = "/dev/vx/dsk/wls90admin/wlsadmin"
    FSType = vxfs
```

```
        FsckOpt = "-y"
)


NIC wlsadminnic
(
     Device = hme0
     NetworkType = ether
)


IP wlsadminip
(
      Device = hme0
      Address = "192.126.5.166"
      NetMask = "255.255.255.0"
)


WebLogic wls_domain (
  Critical = 0
  BEA_HOME = "/Oracle/Middleware"
  WlstScript = "/Oracle/Middleware/wlserver_10.3/common/bin/wlst.sh"
  ListenAddressPort = "wlsadmin:7001"
  ServerName = AdminServer
  ServerRole = Administrative
  WLSUser = weblogic
  WLSPassword = gunSlsIssSvsNspSjmHmiMj
  SecondLevelMonitor = 5
  ServerStartProgram = "/Oracle/Middleware/user_projects/domains/
                        wls_domain/bin/startWebLogic.sh"
  ServerStopProgram = "/Oracle/Middleware/user_projects/domains/
                       wls_domain/bin/stopWebLogic.sh"
  )
 WebLogic wls_managedserver1 (
  Critical = 0
  AdminURL = "t3://wlsadmin:7001"
  BEA_HOME = "/Oracle/Middleware"
  WlstScript = "/Oracle/Middleware/wlserver_10.3/common/bin/wlst.sh"
  ListenAddressPort = "10.209.73.90:7003"
  ServerName = ManagedServer1
  ServerRole = Managed
  WLSUser = weblogic
  WLSPassword = gunSlsIssSvsNspSjmHmiMj
  SecondLevelMonitor = 5
  ServerStartProgram = "/Oracle/Middleware/user_projects/domains/
```

```
                                wls_domain/bin/startManagedWebLogic.sh"
          ServerStopProgram = "/Oracle/Middleware/user_projects/domains/
                                wls_domain/bin/stopManagedWebLogic.sh"
          )


        wls_domain_res requires wls_ip
         wls_managedserver1 requires wls_domain_res
        wls_domain_res requires wlsadmin_mnt
        wlsadmin_mnt requires wlsadmin_dg
        wlsadmin_ip requires wlsadmin_nic
```

Changes introduced in previous releases

This appendix includes the following topics:

■ Changes introduced in previous releases

Changes introduced in previous releases

The enhancements in the previous releases of Veritas agent for WebLogic Server are as follows:

■ The WebLogic9 agent type is changed to WebLogic agent type.

■ The package name is changed from VRTSwls9 to VRTSwls.

■ The agent now supports secure WebLogic Server configurations, that is, WebLogic Server running on secure ports such as, t3s/https.

■ The following new attributes are added:

   ■ nmUser : Node Manager user name

   ■ nmPassword : Node Manager password

   ■ nmHome : Node Manager location

   ■ WsltScript : The exact path of a customized wlst.sh script

   ■ ShutdownTimeout : The timeout interval to pass to the shutdown() API via wlst.sh

■ The following attributes are modified:

   ■ AdminUrl

   ■ ServerStartProgram

- ■ ServerStopProgram

- ■ The following attributes are deprecated:

  - ■ WL_HOME

  - ■ DomainName

- ■ Enhanced the handling of logging when the user specified in the User attribute, is not present in the /etc/passwd file.

- ■ Fixed an issue with the process filtering of WebLogic Managed Server. The agent now filters processes irrespective of the appearance order of "Name" and "management.server" url.
  Fixed issue with the WebLogic9 agent type import for VCS 5.1 using WebLogic9Types51.cmd script. Earlier, on VCS 5.1, while importing the WebLogic type by executing the WebLogic9Types51.cmd script, the AEPTimeout value was set to 0 instead of 1. This caused the agent to execute the `hatype` command during every entry point.

- ■ Fixed issue with the IP address matching while monitoring the WebLogic instance. The WebLogic agent now correctly matches the IP address given in the agent attributes.

- ■ Changed the logging level from TRACE to INFO, while any WebLogic application process is being killed by the agent.

- ■ Added support for AIX 6.1 on pSeries.

- ■ Added support for SUSE Linux Enterprise Server 9.

- ■ Added support for VCS 5.1 on AIX, Linux, and Solaris.

- ■ Added support for VCS One 5.0.

- ■ Added support for VCS One 2.0.

- ■ Added support for Solaris x86 for VCS 4.1 and 5.0.

- ■ Added support for Internationalization (i18n).

- ■ Added command line argument support for ServerStartProgram and ServerStopProgram attributes.

- ■ Added support for WebLogic Server version 10.3.

- ■ Added support for WebLogic Servers without Node Manager based configuration.

- ■ Added the following attributes:

  - ■ DomainDir

  - ■ WL_HOME

- ■ ServerStartProgram

- ■ ServerStopProgram

■ Integrated with the enhanced version of ACC library, that includes numerous fixes for improved functionality.

■ Fixed the default `csh` shell issue. Previously, if the user had set the csh shell as default, the agent was unable to run the `start` command in the background and was unable to redirect the output of the agent functions.

■ Fixed issue that arose with the SecondLevelMonitor attribute when users used the `csh` shell as default. The users previously could not run the second level check if the file specified in the EnvFile attribute contained `csh` syntax. The function failed and reported errors to the cluster engine log.

■ Fixed the negative timeout value that SecondLevelMonitor used when online.

■ Fixed issue that arose during first level monitor check. Previously, if the first level monitor check failed, the agent was unable to bring the resource offline. Instead, the agent reported the resource state as UNKNOWN.

■ Fixed issue that arose due to the format of the ListenAddressPort attribute. Previously, if the format of ListenAddressPort was IPAddress:Port, the agent was unable to interpret the value correctly.

■ Earlier, the Weblogic server started as root even when the User attribute was set to a non-root user. Now, the Weblogic server correctly starts with the user specified in the User attribute.

Index