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

AIX, HP-UX, Linux, Solaris

5.1

March 2009

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

Document version: 5.1.3

Legal Notice

Copyright © 2009 Symantec Corporation. All rights reserved.

Symantec, the Symantec Logo, Veritas and Veritas Storage Foundation are trademarks or registered trademarks of Symantec Corporation or its affiliates in the U.S. and other countries. Other names may be trademarks of their respective owners.

The product described in this document is distributed under licenses restricting its use, copying, distribution, and decompilation/reverse engineering. No part of this document may be reproduced in any form by any means without prior written authorization of Symantec Corporation and its licensors, if any.

THE DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID. SYMANTEC CORPORATION SHALL NOT BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS DOCUMENTATION. THE INFORMATION CONTAINED IN THIS DOCUMENTATION IS SUBJECT TO CHANGE WITHOUT NOTICE.

The Licensed Software and Documentation are deemed to be commercial computer software as defined in FAR 12.212 and subject to restricted rights as defined in FAR Section 52.227-19 "Commercial Computer Software - Restricted Rights" and DFARS 227.7202, "Rights in Commercial Computer Software or Commercial Computer Software Documentation", as applicable, and any successor regulations. Any use, modification, reproduction release, performance, display or disclosure of the Licensed Software and Documentation by the U.S. Government shall be solely in accordance with the terms of this Agreement.

Symantec Corporation
20330 Stevens Creek Blvd.
Cupertino, CA 95014

http://www.symantec.com

Technical Support

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

Symantec's maintenance 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 Web-based support that provides rapid response and up-to-the-minute information
- Upgrade assurance that delivers automatic software upgrade protection
- Global support that is available 24 hours a day, 7 days a week
- Advanced features, including Account Management Services

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

www.symantec.com/techsupp/

Contacting Technical Support

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

www.symantec.com/business/support/assistance_care.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/techsupp/

Customer service

Customer service information is available at the following URL:

www.symantec.com/techsupp/

Customer Service is available to assist with 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 maintenance contracts
- Information about the Symantec Buying Programs
- Advice about Symantec's technical support options
- Nontechnical presales questions
- Issues that are related to CD-ROMs or manuals

Documentation feedback

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

Maintenance agreement resources

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

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

Additional enterprise services

Symantec offers a comprehensive set of services that allow you to maximize your investment in Symantec products and to develop your knowledge, expertise, and global insight, which enable you to manage your business risks proactively.

Enterprise services that are available include the following:

Symantec Early Warning Solutions	These solutions provide early warning of cyber attacks, comprehensive threat analysis, and countermeasures to prevent attacks before they occur.
Managed Security Services	These services remove the burden of managing and monitoring security devices and events, ensuring rapid response to real threats.
Consulting Services	Symantec Consulting Services provide on-site technical expertise from Symantec and its trusted partners. Symantec Consulting Services offer a variety of prepackaged and customizable options that include assessment, design, implementation, monitoring, and management capabilities. Each is focused on establishing and maintaining the integrity and availability of your IT resources.
Educational Services	Educational Services provide a full array of technical training, security education, security certification, and awareness communication programs.

To access more information about Enterprise services, please visit our Web site at the following URL:

www.symantec.com

Select your country or language from the site index.

Contents

Chapter 1

Introducing the Veritas High Availability Agent for WebLogic Server

This chapter includes the following topics:

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://seer.entsupport.symantec.com/docs/282004.htm

What's new in this agent

The enhancements in this release of WebLogic Server agent are as follows:

- 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.

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 ■ HP-UX—VCS 4.1, 5.0 ■ Linux—VCS 4.0, 4.1, 5.0 ■ Solaris—VCS 4.0, 4.1, 5.0
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 on pSeries ■ HP-UX 11i version 2, HP-UX 11i version 3 ■ Red Hat Enterprise Linux 3.0, 4.0, 5.0 on Intel ■ SUSE Linux Enterprise Server 10 ■ Solaris 8, 9, 10 on SPARC and x86 **Note:** The agent supports zones on Solaris 10. **Note:** For Solaris, Symantec recommends applying the latest Solaris operating system patches available from Sun. Visit the Sun web-site for more information.
WebLogic Server	7.0, 8.1, 9.0, 9.1, 9.2, 10.0, 10.3 and all intermediate minor versions of these releases.

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

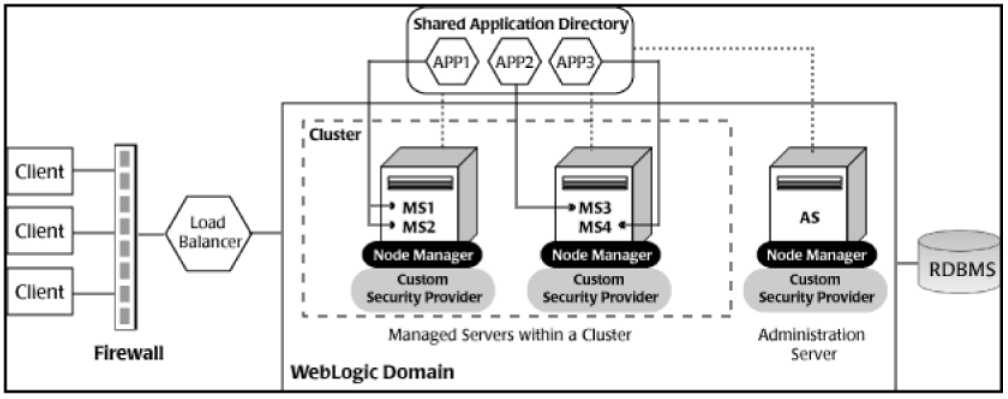
Veritas Cluster Server One	VCS One 2.0 on AIX, HP-UX, Linux, and Solaris

Operating Systems	■ AIX 5.1, 5.2, 5.3 on pSeries ■ HP-UX 11i version 2, HP-UX 11i version 3 ■ Red Hat Enterprise Linux 3.0, 4.0, 5.0 on Intel ■ SUSE Linux Enterprise Server 10 ■ Solaris 8, 9, 10 on SPARC and x86 **Note:** The agent supports zones on Solaris 10. **Note:** For Solaris, Symantec recommends applying the latest Solaris operating system patches available from Sun. Visit the Sun web-site for more information.
WebLogic Server	7.0, 8.1, 9.0, 9.1, 9.2, 10.0, 10.3 and all intermediate minor versions of these releases.

About WebLogic Server

WebLogic Servers fall into two categories: Administrative and Managed. The Administrative Server provides a central point from which you can manage the domain, and it provides access to WebLogic server administration tools [WLS05: *Introduction to 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 till the Administrative server is initialized. For details, refer to description of attributes AdminServerMaxWait and RequireAdminServer.
- For different resource configurations, starts the WebLogic Server component using the mechanism shown as follows.

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

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

Managed server (NNM)	Uses the wlst command `connect`.

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

Clean

The clean function performs the following tasks:

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

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

Chapter 2

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 a WebLogic Server service group.

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, `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 25.
- Install the latest version of ACC Library.
  To install or update the ACC Library package, locate the library and related documentation on the agentpack disc.
  See "Installing the ACC library" on page 22.

For VCS One, do the following:

- Install and configure Veritas Cluster Server One.
  For more information on installing and configuring Veritas Cluster Server refer to, `Veritas Cluster Server One Installation Guide`
- Remove any previous version of this agent.
  To remove the agent,
  See "Removing the agent in a VCS One environment" on page 26.

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.
  You can find the latest version of ACCLib on the agent pack disc at the following location:

Platform	Location
AIX	cd1/platform/application/acc_library/vcs/version_library/pkgs/
HP-UX	cd1/platform/*arch_dist*/application/acc_library/vcs/version_library/pkgs/ where *arch_dist* takes the value 'generic'
Linux	cd1/platform/*arch_dist*/application/acc_library/vcs/version_library/rpms/ where *arch_dist* takes the value 'generic'

Solaris	cd1/platform/*arch_dist*/application/acc_library/vcs/version_library/pkgs/ where *arch_dist* takes the following values: 'sparc' for Solaris SPARC 'sol_x64' for Solaris x64

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

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

  Note: *arch_dist* is not applicable to AIX.

About ACC Library

The operations for the Veritas agent for WebLogic Server depend on a set of Perl modules known as the ACC library. The library must be installed on each system in the cluster that will run the agent for WebLogic Server. The ACC library contains common, reusable functions that perform tasks, such as process identification, logging, and system calls.

Note: If you are installing the agent for WebLogic Server in a VCS 5.0 or VCS One environment, do not install the ACC library package separately. If you are installing the agent in a VCS 4.x environment, you must install the ACC library package before installing the agent.

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. Navigate to the pkgs directory (the pkgs directory on the CD).

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

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

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

4. For HP-UX, install the HP-UX patch PHCO_29042 if it is not already installed.

Installing the agent in a VCS environment

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

To install the agent

1. Log in as superuser.
2. Navigate to the directory containing the package for the platform running in your environment.

AIX	*cd_mount*/aix/application/weblogic_agent/vcs_*version*/*version*_agent/pkgs
HP-UX	*cd_mount*/hpux/generic/application/weblogic_agent/vcs_*version*/*version*_agent/pkgs
Linux	*cd_mount*/linux/generic/application/weblogic_agent/vcs_*version*/*version*_agent/rpms
Solaris	*cd_mount*/solaris/*dist_arch*/application/weblogic_agent/vcs_*version*/*version*_agent/pkgs Where *dist* is the Solaris distribution and *arch* is the Solaris processor architecture.

3. Install the package.

AIX	`# installp -ac -d VRTSwls9.rte.bff VRTSwls9.rte`
HP-UX	`# swinstall -s 'pwd' VRTSwls9`
Linux	`# rpm -ihv \` `VRTSwls9-`*`AgentVersion`*`-GA_GENERIC.noarch.rpm`
Solaris	`# pkgadd -d . VRTSwls9`

About installing the agent in a VCS One environment

You must install the agent for WebLogic Server on all the client systems of the server farm that will host the WebLogic Server service group.

You can install the agent for WebLogic Server using the installagpack program.

Following are the commonly used options that installagpack program supports.

-addtypes	Use this option to add the type definition for the agents that are shipped with agent pack installer. See "Adding the agent type definitions in a VCS One environment" on page 32.
-rmtypes	Use this option to remove the type definition for the agents that are shipped with agent pack installer.
-responsefile	Use this option to perform automated VCS One High Availability Agents Installation using the system and the configuration information that is stored in a specified file instead of prompting for information. The responsefile *response_file* must be a full path name. If -responsefile option is not specified, the response file is automatically generated as installagpack*RANDSTRING*.response, where *RANDSTRING* is a six character string of random alpha-numerals. You must edit the response file to use it for subsequent installations. Variable field definitions are defined within the file.
-rsh	Use this option to specify that rsh and rcp are to be used for communication between systems instead of ssh and scp. This option requires that systems be preconfigured such that rsh commands between systems execute without prompting for passwords or confirmations.

Installing the agent in a VCS One environment

Installing the agent packs involves the following phases:

- Installing the agent packages
- Adding the agent resource type definitions
  See "Adding the agent type definitions in a VCS One environment" on page 32.

Installing the agent packages

You can add the agent packages on one or more client systems of a specific platform type.

To install the Veritas high availability agents

1. Mount the Agent Pack software disc on the client system where you plan to run the installation.
2. Depending on the platform type, navigate to the directory containing the installer for the VCS One agents:

AIX	`cd_mount/aix/high_availability_agents`
HP-UX	`cd_mount/hpux/hpux`*`os_version`*`/high_availability_agents`
Linux	`cd_mount/linux/dist_arch/high_availability_agents` Where dist is the Linux distribution and arch is the architecture.
Solaris	`cd_mount/solaris/dist_arch/high_availability_agents` where *dist* is the distribution and *arch* is the Solaris processor architecture.

3. Start the installagpack program.

```
# ./installagpack
```

4. Enter the name of a client system or client systems where you want to install the agents.
5. Select the agent to install.
6. After the agent is installed, run the installagpack program using the -addtypes option, to add the agent type to the Policy Master configuration.

```
# ./installagpack -addtypes
```

7. Enter the virtual IP address of the Policy Master.
8. Select the agent type you want to add.
9. Review the output as the installation program installs the agent packages.

   You can view installation logs in the /var/VRTS/install/logs directory.

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=WebLogic9
   ```

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

   ```
   # hatype -delete WebLogic9
   ```

   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 VRTSwls9.rte`
HP-UX	`# swremove VRTSwls9`
Linux	`# rpm -e VRTSwls9`
Solaris	`# pkgrm VRTSwls9`

Removing the agent in a VCS One environment

You can remove all the Veritas agent packages that the installagpack program installed, using the uninstallagpack program.

To remove the Veritas high availability agents from client systems

1. Mount the Agent Pack software disc on the client system where you plan to run the uninstallagpack program.
2. Depending on the platform type, navigate to the directory containing the uninstaller for the VCS One agents:

AIX	`cd_mount/aix/high_availability_agents`
HP-UX	`cd_mount/hpux/hpux`*`os_version`*`/high_availability_agents`
Linux	`cd_mount/linux/dist_arch/high_availability_agents` Where dist is the Linux distribution and arch is the architecture.
Solaris	`cd_mount/solaris/`*`dist_arch`*`/high_availability_agents` where *dist* is the distribution and *arch* is the Solaris processor architecture.

3. Start the uninstallagpack program.

```
# ./uninstallagpack
```

4. Enter the names of the client systems from which you want to uninstall the agent. The names must be separated by spaces.
5. Run the installagpack program using the -rmtypes option, to remove the agent type from the Policy Master configuration.

```
./installagpack -rmtypes
```

6. Enter the virtual IP address of the Policy Master.
7. Review the output as the program verifies the agent pack that you installed and removes the agent packages.

   You can view logs in the /var/VRTS/install/logs directory.

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.rte`
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

1. Persistently freeze the service groups that host the application.

   `# hagrp -freeze group -persistent`

2. Stop the cluster services forcibly.

   `# hastop -all -force`

3. Ensure that the agent operations are stopped on all the nodes.

   `# ps -ef |grep WebLogic9`

4. Uninstall the agent package from all the nodes.

   See "Removing the agent in a VCS environment" on page 25.

5. Install the new agent on all the nodes.

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

6. Copy the new WebLogic9Types.cf file from the agent's sample conf directory,

VCS 4.x	/etc/VRTSagents/ha/bin/WebLogic9
VCS 5.0	/etc/VRTSvcs/conf/sample_WebLogic9

   to the VCS conf directory /etc/VRTSvcs/conf/config.

Note: If you are using Solaris Zones, copy the WebLogic9Types_zones.cf file from the agent's sample conf directory.

---

7 Check for the changes in the resource values required, if any, due to the new agent types file.

Note: To note the list of changed attributes, compare the new type definition file with the old type definition file.

8 Start the cluster services.

```
# hastart
```

9 Start the agent on all nodes, if not started.

```
# haagent -start WebLogic9 -sys System
```

10 Unfreeze the service groups once all the resources come to an online steady state.

```
# hagrp -unfreeze group -persistent
```

Upgrading the agent in a VCS One environment

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

1 Freeze service groups that hosts the application.

```
# hagrp -freeze -propagate group
```

2 Stop the clients forcibly. Execute the following command from the Policy Master.

```
# hastop -client -sys vcslx295 -force
```

3 Ensure that the agent operations are stopped on all the nodes.

```
# ps -ef |grep WebLogic9
```

4 Uninstall the agent package from all the nodes.

See "Removing the agent in a VCS One environment" on page 26.

5 Install the new agent on all the nodes in the cluster.

See "Installing the agent in a VCS One environment" on page 24.

6 Add the agent types, using the installagpack program.

See "Adding the agent type definitions in a VCS One environment" on page 32.

7 Check for the changes in the resource values required, if any, due to the new agent types file.

8 Start the clients.

```
# hastart -client
```

9 Start the agent on all nodes, if not started.

```
# haagent -start WebLogic9 -sys System
```

10 Unfreeze the service groups.

```
# hagrp -unfreeze -propagate group
```

Chapter 3

Preparing to configure 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 a WebLogic Server resource. 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 65.

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 4.x	/etc/VRTSvcs/conf/sample_WebLogic9/WebLogic9Types.cf
VCS 5.0	/etc/VRTSagents/ha/conf/WebLogic9/WebLogic9Types.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*.

Adding the agent type definitions in a VCS One environment

For VCS One, you must add the agent type definitions to the Policy Master database configuration. You can perform this task on the Policy Master (PM) system or from any other client system in the server farm.

Note: You must add the agent resource type definitions only one time per platform type.

To add the Veritas high availability agent resource types to the PM database configuration

1 If you plan to add the resource type definitions from the client system where you ran the installer, then you must set up rsh or paswordless ssh communications between this client system and the PM system.

For information on configuring ssh for remote communication, refer to *Veritas Cluster Server One Installation Guide*.

2 Make sure that the PM daemon is running. Depending on the system you choose to add the resource types, run the following command:

From any client system in the server farm	`# haclus -display` The output should show ClusterState is RUNNING.
From the Policy Master system	`# haadmin -state` The output should show the PMSG is ONLINE on one node, OFFLINE on the other.

3 Perform the following steps only if you plan to run the installation program on the Policy Master system:

- Mount the Agent Pack software disc.
- Depending on the platform type, navigate to the directory containing the installer for the agents:

AIX	`cd aix/high_availability_agents`
HP-UX	`cd hpux/hpux<os_version>/high_availability_agents`
Linux	`cd linux/dist_arch/high_availability_agents` Where dist is the Linux distribution and arch is the architecture.
Solaris	`cd solaris/dist_arch/high_availability_agents` where *dist* is distribution and *arch* is the Solaris processor architecture.

4 Enter the command to start the agent pack installer for adding resource types to the Policy Master configuration database. Use the `-addtypes` flag:

```
# ./installagpack -addtypes
```

Note: The -addtypes option must be run from the client for which you want to add resource types. Depending on the platform type, navigate to the directory containing the agent pack installer.The agent pack installer determines the client platform and adds types specific to that platform.

5 When the installer prompts, enter the virtual IP address of the Policy Master.

6 If you are running the command from a client system, then review the output as the installer verifies communication with the Policy Master system.

7 Review the output as the installer adds the agent types to the PM database configuration and copies the appropriates types.xml files to the PM system.

You can view installation logs in the /var/VRTS/install/logs directory.

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
DomainName	The name of the WebLogic Server domain to which the instance belongs. The WebLogic Server uses this attribute to connect to the Node Manager using the wlst.sh utility. Specify this attribute for Administrative and Managed Servers. If the SecondLevelMonitor attribute is specified, specify this attribute for the Node Manager also. See "Uniquely identifying WebLogic Server instances" on page 42. Type and dimension: string-scalar Default: "" Example: WLS90Domain
ListenAddressPort	The Listen Address and port of the WebLogic instance. The format is ListenAddress:port. Ensure that the ListenAddress string resolves to the proper IP Address, using the network name service that you used on the host. The WebLogic Server connects to the ListenAddress on the specified port through the wlst.sh API. Specify this attribute for Administrative and Managed Servers only. Type and dimension: string-scalar Default: "" Example: wls90adminsol.veritas.com:7001 or wls90adminsol.veritas.com:5556

Table 3-1 Required attributes *(continued)*

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

Table 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 42. Type and dimension: string-scalar Default: "" Example: AdminServer
WL_HOME	The absolute path to the product installation directory of the WebLogic Server. The agent for WebLogic Server uses this attribute to locate the wlst.sh utility and the Node Manager home directory. Type and dimension: string-scalar Default: "" Example: /bea/wls90/admin/weblogic90
WLSUser	The user name of the user that is connecting the wlst.sh utility to the server running the WebLogic Server instance, along with WLSPassword. Type and dimension: string-scalar Default: ""
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

Table 3-1 Required attributes *(continued)*

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

Table 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: ServerStopProgram = "/wls/my_domain/stopManagedWebLogic.sh Managed1 t3://adminurl:7001 weblogic passwd" If no arguments are passed (for example, ServerStopProgram = "/wls/my_domain/stopManagedWebLogic.sh", the agent forms the command line as follows: `$ServerStopProgram $ServerName $AdminURL $WLSUser $WLSPassword` Type and dimension: string-scalar Default: "" Example: /bea/user_projects/domains/WLS90Domain/bin/stopManagedWebLogic.sh

Table 3-2 lists the optional attributes.

Table 3-2 Optional attributes

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

Table 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 49. 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 42. 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. 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 VRTSWebLogic9Key.properties and VRTSWebLogic9Config.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 66. 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. **Note:** Exercise caution while setting SecondLevelMonitor to large numbers. For example, if the MonitorInterval is set to 60 seconds and the SecondLevelMonitor is set to 100, then wlst.sh is executed every 100 minutes, which may not be as often as intended. For maximum flexibility, no upper limit is defined for SecondLevelMonitor. Type and dimension: integer-scalar Default: 0 Example: 1

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 DomainName must be unique for the Administrative Server instance.
- To uniquely identify a Managed Server instance, do the following:
  - The combination of ServerName and DomainName must be unique for the Managed Server instance.
  - The value of the AdminUrl attribute must match the value of management server that appears in the long listing of processes for the Managed Server instance.
- To uniquely identify a Node Manager instance, the value of the nmListenAddressPort attribute must match the values of ListenAddress and ListenPort that appear in the long listing of processes for the Node Manager instance.

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.

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	-	-	-	Yes
BEA_HOME	Yes	Yes	Yes	Yes
WL_HOME	Yes	Yes	Yes	Yes
DomainName	-	Yes	Yes	Yes
DomainDir	-	Yes	Yes	Yes
ListenAddressPort	-	-	Yes	Yes
MonitorProgram	Opt	Opt	Opt	Opt
nmListenAddressPort	Yes	Yes	Yes	Yes
nmType	Yes	Yes	Yes	Yes
ServerName	-	-	Yes	Yes
ServerRole	Yes	Yes	Yes	Yes
WLSUser	Yes	Yes	Yes	Yes
WLSPassword	Yes	Yes	Yes	Yes
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	-	-
BEA_HOME	Yes	Yes	Yes	Yes
WL_HOME	Yes	Yes	Yes	Yes
DomainName	Yes	Yes	Yes	Yes
DomainDir	Yes	Yes	Yes	Yes
ListenAddressPort	Yes	Yes	Yes	Yes
MonitorProgram	Opt	Opt	Opt	Opt
nmListenAddressPort	-	-	-	-
nmType	-	-	-	-
ServerName	Yes	Yes	Yes	Yes
ServerRole	Yes	Yes	Yes	Yes
WLSUser	-	Yes	-	Yes
WLSPassword	-	Yes	-	Yes
RequireAdminServer	Yes	Yes	-	-
AdminServerMaxWait	Yes	Yes	-	-
SecondLevelMonitor	0	> 0	0	> 0
ServerStartProgram	Yes	Yes	Yes	Yes
ServerStopProgram	Yes	Yes	Yes	Yes
User	Yes	Yes	Yes	Yes

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:

- See "Node Manager without SLM enabled" on page 71.

- See "Node Manager with SLM enabled" on page 72.
- See "Administrative Server (NM) without SLM enabled" on page 73.
- See "Administrative Server (NM) with SLM enabled" on page 74.
- See "Managed Server (NM) without SLM enabled" on page 75.
- See "Managed Server (NM) with SLM enabled" on page 76.
- See "Managed Server (NNM) without SLM enabled" on page 77.
- See "Managed Server (NNM) with SLM enabled" on page 78.
- See "Administrative Server (NNM) without SLM enabled" on page 79.
- See "Administrative Server (NNM) with SLM enabled" on page 80.

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 WebLogic9 resource that uses WebLogic provided scripts to start and stop the WebLogic server it is recommended to have the boot identity files to avoid storing unencrypted credentials in startup/shutdown scripts. The boot identity file boot.properties should be created for the WebLogic server and placed in the security directory of the server.

For more details, refer to http://edocs.bea.com/wls/docs90/server_start/overview.html#1068976.

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

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:

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.

Chapter 4

Configuring the service groups for WebLogic Server

This chapter includes the following topics:

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 69.

Chapter 5

Troubleshooting the agent for WebLogic Server

This chapter includes the following topics:

Using 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 13.

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 a WebLogic Server resource, ensure that you configure the resource 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
nmListenAddressPort	wls90admsol:5556
nmType	ssl
ServerRole	NodeManager
WL_HOME	/bea/wls90/admin/weblogic90

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

   ```
   # su - weblogic
   ```

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

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

5. Start the WebLogic Server Scripting Tool:

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

6 Start the Node Manager:

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

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

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

7 Enter this command:

```
# exit()
```

If the Node Manager works properly outside the 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
DomainName	WLS90Domain
nmListenAddressPort	wls90admsol:5556
WL_HOME	/bea/wls90/admin/weblogic90
DomainDir	/bea/wls90/admin/user_projects/domains/WLS90Domain
nmType	ssl
User	weblogic

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

```
# su - weblogic
```

4 Go to the directory specified in the BEA_HOME attribute:

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

5 Start the WebLogic Server Scripting Tool:

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

6 Connect to the Node Manager:

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

7 Start the Administrative Server:

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

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

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

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

Reviewing error log files

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

Using WebLogic Server log files

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

- For Node Managers:

  WL_HOME/common/nodemanager/nodemanager.log

- For Administrative Servers:

```
DomainDir/servers/ServerName/ServerName.log
DomainDir/servers/ServerName/ServerName.out
```

- For Managed Servers:

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

Reviewing 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.
- The VCS One engine log file is /var/VRTSvcsone/log/engine_A.log.
- The VCS One client log file is /var/VRTSvcsone/log/vcsoneclientd_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/.VRTSWebLogic9/WLS90Mng01_nodemanager_online.out
/tmp/.VRTSWebLogic9/WLS90Mng01_nodemanager_offline.out
/tmp/.VRTSWebLogic9/WLS90Mng01_nodemanager_clean.out
/tmp/.VRTSWebLogic9/WLS90Mng01_nodemanager_monitor.out
```

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

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

Problems starting a Managed Server through the administrative console

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

If the Node Manager is running on a virtual host, this communication may fail. This failure may occur because the Node Manager uses default SSL certificates that contain the real host name of the physical node on which the Node Manager is running. The URL used for connecting to the Node Manager contains the virtual host name of the Node Manager, which is different from the physical host name of the node. The Administrative Server rejects the communication because of this mismatch.

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

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

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

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

  BEA Systems recommends generating new SSL certificates using reliable certification authorities as best security practice. Otherwise, you can generate certificates and keystores which use virtual hostname, using the tools, CertGen and ImportPrivateKey that WebLogic provides.
- Disable the host name verification function
  You can disable the host name verification function in the Administrative Server properties. For details about disabling the function, refer to the following link:
  http://e-docs.bea.com/wls/docs90/ConsoleHelp/taskhelp/security/DisableHostNameVerification.html.

Unable to bring two or more VCS resources offline simultaneously

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

See “Editing the WebLogic stop script” on page 47.

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://e-docs.bea.com/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
```

Appendix A

Sample Configurations

This appendix includes the following topics:

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 ./VRTSWebLogic9Config.properties\
   -userkeyfile ./VRTSWebLogic9Key.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
   ./VRTSWebLogic9Config.properties \ -userkeyfile
   ./VRTSWebLogic9Key.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 ./VRTSWebLogic9Config.properties\
-userkeyfile ./VRTSWebLogic9Key.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 ./VRTSWebLogic9Config.properties\
-userkeyfile ./VRTSWebLogic9Key.properties GETSTATE
Current state of "AdminServer" : RUNNING
```

Sample agent type definition for WebLogic server

Examples of agent type definition files follow.

For VCS 4.x

After importing the agent types into the cluster, if you save the configuration on your system disk using the `haconf -dump` command, you can find the WebLogic9Types.cf file in the /etc/VRTSvcs/conf/config cluster configuration directory.

An excerpt from this file follows.

```
type WebLogic9 (
    static str ArgList[] = { ResLogLevel, State, IState, AdminURL,
                             BEA_HOME, WL_HOME, DomainName, DomainDir,
                             ListenAddressPort, MonitorProgram,
                             nmListenAddressPort, nmType, ServerName,
                             ServerRole, User, WLSUser, WLSPassword,
                             ServerStartProgram, ServerStopProgram,
                             RequireAdminServer, AdminServerMaxWait,
                             SecondLevelMonitor }
    str ResLogLevel = INFO
    str AdminURL
    str BEA_HOME
    str WL_HOME
```

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

For VCS 5.0

After importing the agent types into the cluster, if you save the configuration on your system disk using the `haconf -dump` command, you can find the WebLogic9Types.cf file in the /etc/VRTSagents/ha/conf/config cluster configuration directory.

An excerpt from this file follows.

```
type WebLogic9 (
    static str AgentFile = "/opt/VRTSvcs/bin/Script50Agent"
    static str AgentDirectory = "/opt/VRTSagents/ha/bin/WebLogic9"
    static str ArgList[] = { ResLogLevel, State, IState, AdminURL,
                             BEA_HOME, WL_HOME, DomainName, DomainDir,
                             ListenAddressPort, MonitorProgram,
                             nmListenAddressPort, nmType, ServerName,
                             ServerRole, User, WLSUser, WLSPassword,
                             ServerStartProgram, ServerStopProgram,
                             RequireAdminServer, AdminServerMaxWait,
                             SecondLevelMonitor }
    str ResLogLevel = INFO
    str AdminURL
    str BEA_HOME
    str WL_HOME
    str DomainName
    str DomainDir
    str ListenAddressPort
```

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

For VCS One

After installing the agent, go to the /etc/VRTSagents/ha/conf/WebLogic9/ directory to view the WebLogic9Types.*platform*.xml agent definition file.

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

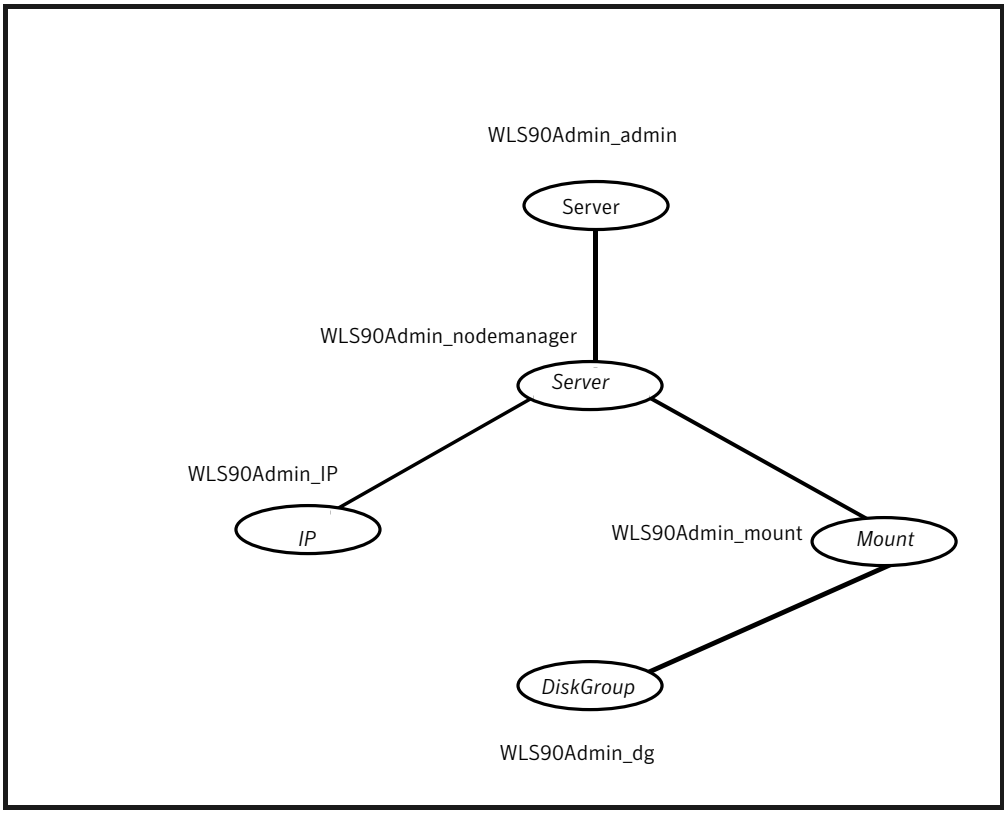
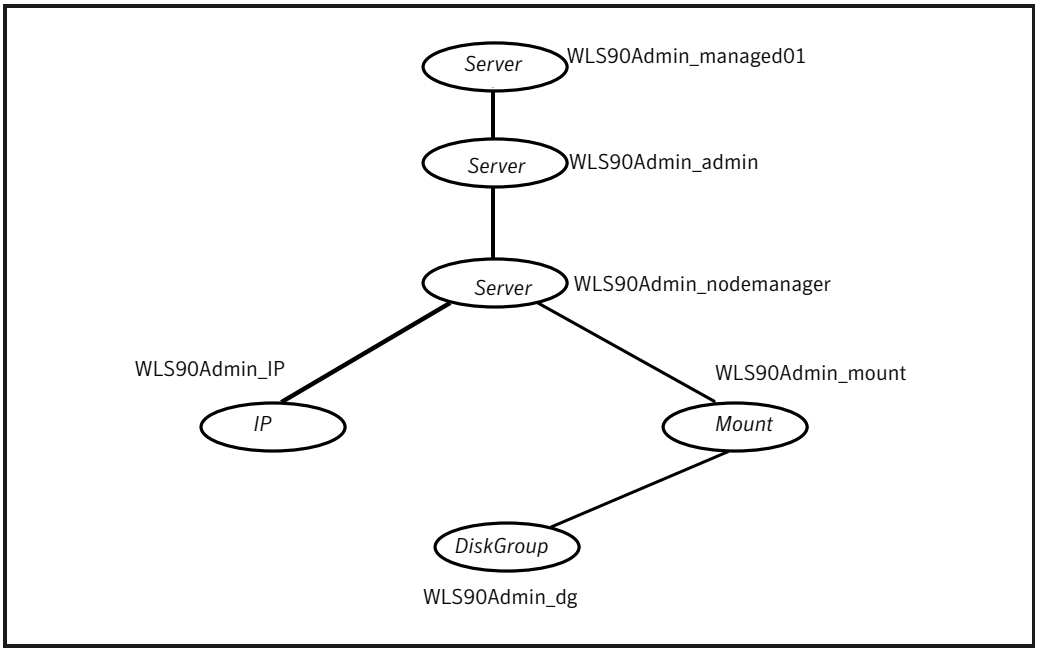


Figure A-2 shows a service group with Administrative and Managed Servers.

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	/bea/wls90/admin/wls91

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

Attribute	Value
WL_HOME	/bea/wls90/admin/wls91/weblogic91
DomainName	
ListenAddressPort	wls90admsol:5556
MonitorProgram	
nmListenAddressPort	wls90admsol:5556
nmType	ssl
ServerName	
ServerRole	NodeManager
User	root
WLSUser	weblogic
WLSPassword	EQFsHqkkMNRkL
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
AdminURL	
BEA_HOME	/bea/wls90/admin/wls91

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

Attribute	Value
WL_HOME	/bea/wls90/admin/wls91/weblogic91
DomainName	WLS91Domain
DomainDir	/bea/wls90/admin/wls91/user_projects/domains/WLS91Domain
ListenAddressPort	wls90admsol:5556
MonitorProgram	
nmListenAddressPort	wls90admsol:5556
nmType	ssl
ServerName	
ServerRole	NodeManager
User	root
WLSUser	weblogic
WLSPassword	EQFsHqkkMNRkL
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	

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

Attribute	Value
BEA_HOME	/bea/wls90/admin/wls91
WL_HOME	/bea/wls90/admin/wls91/weblogic91
DomainName	WLS91Domain
DomainDir	/bea/wls90/admin/wls91/user_projects/domains/WLS91Domain
ListenAddressPort	wls90admsol:7011
MonitorProgram	
nmListenAddressPort	wls90admsol:5556
nmType	ssl
ServerName	AdminServer
ServerRole	Administrative
User	root
WLSUser	weblogic
WLSPassword	EQFsHqkkMNRkL
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

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

Attribute	Value
AdminURL	
BEA_HOME	/bea/wls90/admin/wls91
WL_HOME	/bea/wls90/admin/wls91/weblogic91
DomainName	WLS91Domain
DomainDir	/bea/wls90/admin/wls91/user_projects/domains/WLS91Domain
ListenAddressPort	wls90admsol:7011
MonitorProgram	
nmListenAddressPort	wls90admsol:5556
nmType	ssl
ServerName	AdminServer
ServerRole	Administrative
User	root
WLSUser	weblogic
WLSPassword	EQFsHqkkMNRkL
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://wls90admsol:7011
BEA_HOME	/bea/wls90/admin/wls91
WL_HOME	/bea/wls90/admin/wls91/weblogic91
DomainName	WLS91Domain
DomainDir	/bea/wls90/admin/wls91/user_projects/domains/WLS91Domain
ListenAddressPort	wls90admsol:7012
MonitorProgram	
nmListenAddressPort	wls90admsol:5556
nmType	ssl
ServerName	ManagedServer01
ServerRole	Managed
User	root
WLSUser	weblogic
WLSPassword	EQFsHqkkMNRkL
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://wls90admsol:7011
BEA_HOME	/bea/wls90/admin/wls91
WL_HOME	/bea/wls90/admin/wls91/weblogic91
DomainName	WLS91Domain
DomainDir	/bea/wls90/admin/wls91/user_projects/domains/WLS91Domain
ListenAddressPort	wls90admsol:7012
MonitorProgram	
nmListenAddressPort	wls90admsol:5556
nmType	ssl
ServerName	ManagedServer01
ServerRole	Managed
User	root
WLSUser	weblogic
WLSPassword	EQFsHqkkMNRkL
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	http://wls90admsol:7011
BEA_HOME	/bea/wls90/admin/wls91
WL_HOME	/bea/wls90/admin/wls91/weblogic91
DomainName	WLS91Domain
DomainDir	/bea/wls90/admin/wls91/user_projects/domains/WLS91Domain
ListenAddressPort	wls90admsol:7012
MonitorProgram	
nmListenAddressPort	
nmType	ssl
ServerName	ManagedServer01
ServerRole	Managed
User	root
WLSUser	
WLSPassword	
ServerStartProgram	/bea/wls90/admin/wls91/user_projects/domains/WLS91Domain/bin/startManagedWebLogic.sh
ServerStopProgram	/bea/wls90/admin/wls91/user_projects/domains/WLS91Domain/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	
BEA_HOME	/bea/wls90/admin/wls91
WL_HOME	/bea/wls90/admin/wls91/weblogic91
DomainName	WLS91Domain
DomainDir	/bea/wls90/admin/wls91/user_projects/domains/WLS91Domain
ListenAddressPort	wls90admsol:7011
MonitorProgram	
nmListenAddressPort	
nmType	ssl
ServerName	ManagedServer01
ServerRole	Managed
User	root
WLSUser	weblogic
WLSPassword	EQFsHqkkMNRkL
ServerStartProgram	/bea/wls90/admin/wls91/user_projects/domains/WLS91Domain/bin/startManagedWebLogic.sh
ServerStopProgram	/bea/wls90/admin/wls91/user_projects/domains/WLS91Domain/bin/stopManagedWebLogic.sh
RequireAdminServer	false
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	/bea/wls90/admin/wls91
WL_HOME	/bea/wls90/admin/wls91/weblogic91
DomainName	WLS91Domain
DomainDir	/bea/wls90/admin/wls91/user_projects/domains/WLS91Domain
ListenAddressPort	wls90admsol:7011
MonitorProgram	
nmListenAddressPort	
nmType	ssl
ServerName	AdminServer
ServerRole	Administrative
User	root
WLSUser	weblogic
WLSPassword	EQFsHqkkMNRkL
ServerStartProgram	/bea/wls90/admin/wls91/user_projects/domains/WLS91Domain/bin/startWebLogic.sh
ServerStopProgram	/bea/wls90/admin/wls91/user_projects/domains/WLS91Domain/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	http://wls90adminsol:7011
BEA_HOME	/bea/wls90/admin/wls91
WL_HOME	/bea/wls90/admin/wls91/weblogic91
DomainName	WLS91Domain
DomainDir	/bea/wls90/admin/wls91/user_projects/domains/WLS91Domain
ListenAddressPort	wls90admsol:7012
MonitorProgram	
nmListenAddressPort	
nmType	ssl
ServerName	AdminServer
ServerRole	Administrative
User	root
WLSUser	weblogic
WLSPassword	EQFsHqkkMNRkL
ServerStartProgram	/bea/wls90/admin/wls91/user_projects/domains/WLS91Domain/bin/startWebLogic.sh
ServerStopProgram	/bea/wls90/admin/wls91/user_projects/domains/WLS91Domain/bin/stopWebLogic.sh
RequireAdminServer	false
AdminServerMaxWait	15
SecondLevelMonitor	1

Service group dependencies for WebLogic Server

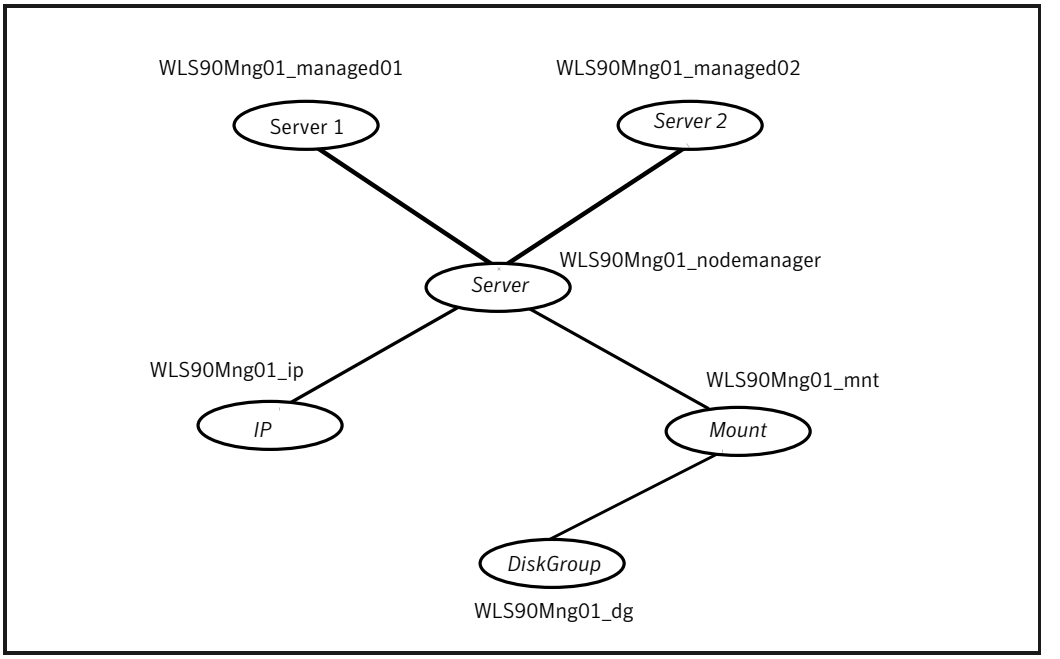
Cluster administrators use Service Group dependencies to create links between unrelated Service Group objects within a cluster. In this version of WebLogic Server, you no longer require Service Group dependencies.

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 49.

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 wls90Admin
(
SystemList = { systemA = 1, systemB = 2 }
)

DiskGroup wls90Admin_dg
(
```

```
    DiskGroup = wls90admin
)

Mount wls90Admin_mnt
(
    MountPoint = "/wls90/admin"
    BlockDevice = "/dev/vx/dsk/wls90admin/wlsadmin"
    FSType = vxfs
    FsckOpt = "-y"
)

NIC wls90Admin_nic
(
    Device = hme0
    NetworkType = ether
)

IP wls90Admin_ip
(
     Device = hme0
     Address = "192.126.5.166"
     NetMask = "255.255.255.0"
)

WebLogic9 WLS90Admin_admin
(
    Critical = 0
    BEA_HOME = "/bea/wls90/admin"
    WL_HOME = "/bea/wls90/admin/weblogic90"
    DomainName = WLS90Domain
    DomainDir ="/bea/wls90/admin/user_projects/domains/WLS90Domain"
    ListenAddressPort = "wls90admhp:7001"
    nmListenAddressPort = "wls90admhp:5556"
    nmType = ssl
    ServerName = AdminServer
    ServerRole = Administrative
    User = weblogic
    WLSUser = weblogic
    WLSPassword = HTIvKTlTNnINjNKnL
    SecondLevelMonitor = 3
)

wls90Admin_app requires wls90Admin_ip
```

```
wls90Admin_app requires wls90Admin_mnt
wls90Admin_ip  requires wls90Admin_nic
wls90Admin_mnt requires wls90Admin_dg
```

Sample configuration in a VCS One environment

To view a sample VCS One configuration file (main.xml) with an Administrative Server instance, a Node Manager instance, and a Managed Server instance, go to the /etc/VRTSagents/ha/conf/WebLogic9/ directory.

Index

T

U

V

W