

# Veritas™ High Availability Agent for WebSphere Application Server Installation and Configuration Guide

AIX, HP-UX, Linux, Solaris

5.1

# Veritas High Availability Agent for WebSphere Application 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: 5.1.8.0

Document version: 5.1.8.0.2

## Legal Notice

Copyright © 2011 Symantec Corporation. All rights reserved.

Symantec, the Symantec logo, Veritas, Veritas Storage Foundation, CommandCentral, NetBackup, Enterprise Vault, and LiveUpdate are trademarks or registered trademarks of Symantec corporation or its affiliates in the U.S. and other countries. Other names may be trademarks of their respective owners.

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

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

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

Symantec Corporation  
350 Ellis Street  
Mountain View, CA 94043  
<http://www.symantec.com>

# Technical Support

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

Symantec's support offerings include the following:

- A range of support options that give you the flexibility to select the right amount of service for any size organization
- Telephone and/or Web-based support that provides rapid response and up-to-the-minute information
- Upgrade assurance that delivers software upgrades
- Global support purchased on a regional business hours or 24 hours a day, 7 days a week basis
- Premium service offerings that include Account Management Services

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

[www.symantec.com/business/support/index.jsp](http://www.symantec.com/business/support/index.jsp)

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

## Contacting Technical Support

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

[www.symantec.com/business/support/contact\\_techsupp\\_static.jsp](http://www.symantec.com/business/support/contact_techsupp_static.jsp)

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

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

- Product release level

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

## Licensing and registration

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

[www.symantec.com/business/support/](http://www.symantec.com/business/support/)

## Customer service

Customer service information is available at the following URL:

[www.symantec.com/business/support/](http://www.symantec.com/business/support/)

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

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

## Support agreement resources

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

Asia-Pacific and Japan	<a href="mailto:customercare_apac@symantec.com">customercare_apac@symantec.com</a>
Europe, Middle-East, and Africa	<a href="mailto:semea@symantec.com">semea@symantec.com</a>
North America and Latin America	<a href="mailto:supportsolutions@symantec.com">supportsolutions@symantec.com</a>

## Documentation

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

<https://sort.symantec.com/documents>

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

[doc\\_feedback@symantec.com](mailto:doc_feedback@symantec.com)

## About Symantec Connect

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

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

# Contents

Technical Support .....	4
Chapter 1      Introducing the Veritas High Availability Agent for WebSphere Application Server .....	9
About the Veritas agent for WebSphere Application Server .....	9
What's new in this agent .....	10
Supported software .....	10
WebSphere Application Server agent functions .....	11
Online .....	11
Offline .....	12
Monitor .....	12
Clean .....	13
Chapter 2      Installing, upgrading, and removing the agent for WebSphere Application Server .....	15
Before you install the Veritas agent for WebSphere Application Server .....	15
Prerequisites for enabling i18n support .....	16
About the ACC library .....	17
Installing the ACC library .....	17
Installing the agent in a VCS environment .....	18
Removing the agent in a VCS environment .....	19
Removing the ACC library .....	20
Updating the WebSphere agent to latest release .....	21
Chapter 3      Configuring the agent for WebSphere Application Server .....	23
About configuring the Veritas agent for WebSphere Application Server .....	23
Importing the agent types files in a VCS environment .....	24
WebSphere Application Server agent attributes .....	26
Uniquely identifying WebSphere Application Server instances .....	29
Important considerations while configuring the agent .....	30
Service group configuration options .....	31

Chapter 4	Configuring the service groups for WebSphere Application Server .....	35
	Before configuring the service groups for WebSphere Application Server .....	35
	Configuring service groups for WebSphere Application Server .....	36
	Allocating shared disk resource for the WebSphere node .....	36
	Creating a Veritas disk group, volume, and file system .....	36
	Obtaining dedicated virtual IP addresses and host names .....	36
	Obtaining a dedicated user account if needed .....	37
	Creating service group and supporting resources .....	37
	Installing the WebSphere software .....	37
	Placing the WebSphere Application Server under cluster control .....	39
Chapter 5	Troubleshooting the agent for WebSphere Application Server .....	41
	Using the correct software and operating system versions .....	41
	Meeting prerequisites .....	42
	Configuring WebSphere Application Server resources .....	42
	Starting the WebSphere Application Server instance outside a cluster .....	42
	Reviewing error log files .....	45
	Reviewing cluster log files .....	45
	Reviewing agent log files .....	45
	Using trace level logging .....	45
	Defining additional environment variables for a WebSphere Application Server instance .....	46
Appendix A	Sample Configurations .....	49
	About sample configurations for the agent for WebSphere Application Server .....	49
	Sample agent type definition for WebSphere Application Server .....	49
	Sample configuration in a VCS environment .....	51
Appendix B	Changes introduced in previous releases .....	57
	Changes introduced in previous releases .....	57
Index	.....	59



# Introducing the Veritas High Availability Agent for WebSphere Application Server

This chapter includes the following topics:

- [About the Veritas agent for WebSphere Application Server](#)
- [What's new in this agent](#)
- [Supported software](#)
- [WebSphere Application Server agent functions](#)

## About the Veritas agent for WebSphere Application 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 WebSphere Application Server provides high availability for WebSphere Application Server in a clustered environment.

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

- Deployment Manager
- Node Agent

■ Application Server

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

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

## What's new in this agent

The enhancements in this release of WebSphere agent are as follows:

- Added support for AIX 7.1 and Red Hat Enterprise Linux 6.0.

For information on the changes introduced in the previous releases:

See “[Changes introduced in previous releases](#)” on page 57.

## Supported software

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

Veritas Cluster Server

- AIX–VCS 4.0, 5.0, 5.1
- HP-UX–VCS 4.1, 5.0
- Linux–VCS 4.0, 4.1, 5.0, 5.1
- Solaris–VCS 4.0, 4.1, 5.0, 5.1

and all intermediate Maintenance Packs of these releases.

ACC Library

5.1 and later

Review the ACC Library version for i18n support.

See “[Prerequisites for enabling i18n support](#)” on page 16.

Operating Systems	<ul style="list-style-type: none"> <li>■ AIX 5.1, 5.2, 5.3, 6.1, 7.1 on pSeries</li> <li>■ HP-UX 11i v2, HP-UX 11i v3 on Itanium and PA-RISC</li> <li>■ Red Hat Enterprise Linux (RHEL) 3, 4, 5 on Intel and ppc64 RHEL 6.0 on Intel</li> <li>■ SUSE Linux Enterprise Server 9, 10, 11 on Intel and ppc64</li> <li>■ Solaris 8, 9, 10 on SPARC and x86</li> </ul> <p><b>Note:</b> The agent supports zones on Solaris in both VCS environments.</p>
WebSphere Application Server	<p>5.0, 5.1, 6.0, 6.1, 7.0</p> <p>and all intermediate minor versions of these releases.</p>

## WebSphere Application Server agent functions

The agent consists of resource type declarations and agent executables. The agent executables are organized into online, offline, monitor, and clean functions.

### Online

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

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

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

## Offline

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

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

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

## Monitor

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

The monitor function performs the following tasks:

- First-level monitoring quickly checks for the existence of the system process (the Java Virtual Machine) that represents the WebSphere Application Server instance. It determines the process existence by scanning the system process table and searching for strings in the process command line that uniquely identify the JVM process associated with the WebSphere Application Server instance. These search strings include the values specified in resource attributes `WAS_HOME`, `WAS_NODE`, and `ServerName`.
- If second-level monitoring is enabled (if `SecondLevelMonitor > 0`), the monitor function performs a deeper, more thorough state check of the WebSphere Application Server. Second-level monitoring uses the IBM-supplied utility program `serverStatus.sh`. The output from this program is parsed to confirm the server is running.

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

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

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

This feature allows VCS administrator to define custom programs that determine the state of the WebSphere Application Server. For example, the administrator may want to test the status of a J2EE component running inside the server and ensure that the underlying application is functioning properly.

See [“WebSphere Application Server agent attributes”](#) on page 26.

## Clean

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

The clean function performs the following tasks:

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



# Installing, upgrading, and removing the agent for WebSphere Application Server

This chapter includes the following topics:

- [Before you install the Veritas agent for WebSphere Application Server](#)
- [About the ACC library](#)
- [Installing the ACC library](#)
- [Installing the agent in a VCS environment](#)
- [Removing the agent in a VCS environment](#)
- [Removing the ACC library](#)
- [Updating the WebSphere agent to latest release](#)

## Before you install the Veritas agent for WebSphere Application Server

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

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

For VCS, do the following:

- Install and configure Veritas Cluster Server.  
For more information on installing and configuring Veritas Cluster Server, refer to the *Veritas Cluster Server Installation Guide*.
- Remove any previous version of this agent.  
To remove the agent,  
See “[Removing the agent in a VCS environment](#)” on page 19.
- Install the latest version of ACC Library.  
To install or update the ACC Library package, locate the library and related documentation in the agentpack tarball.  
See “[Installing the ACC library](#)” on page 17.

## Prerequisites for enabling i18n support

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

- Install ACCLib version 5.1.2.0 or later.  
See “[Installing the ACC library](#)” on page 17.
- For VCS 5.0 and earlier releases, copy the latest ag\_i18n\_inc.pm module from the following location on the agent pack disc.

---

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

---

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

where *arch\_dist* takes the following values:  
'sol\_sparc' for Solaris SPARC  
'sol\_x64' for Solaris x64  
'generic' for HP-UX and Linux

---

**Note:** *arch\_dist* is not applicable to AIX.

---



## About the ACC library

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

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

## Installing the ACC library

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

### To install the ACC library

- 1 Log in as superuser.
- 2 Download ACC Library.

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

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

AIX	<code>cd1/aix/vcs/application/acc_library/version_library/pkgs</code>
HP-UX	<code>cd1/hpux/generic/vcs/application/acc_library/version_library/pkgs</code>
Linux	<code>cd1/linux/generic/vcs/application/acc_library/version_library/rpms</code>
Solaris	<code>cd1/solaris/dist_arch/vcs/application/acc_library/version_library/pkgs</code>

where *dist\_arch* is `sol_sparc` or `sol_x64`.

- 4 If you downloaded the individual ACCLib tarball, navigate to the pkgs directory (for AIX, HP-UX, and Solaris), or rpms directory (for Linux).
- 5 Install the package. Enter **Yes** if asked to confirm overwriting of files in the existing package.

```
AIX          # installp -ac -d VRTSacclib.bff VRTSacclib

HP-UX        # swinstall -s 'pwd' VRTSacclib

Linux        # rpm -i \
              VRTSacclib-VersionNumber-GA_GENERIC.noarch.rpm

Solaris      # pkgadd -d VRTSacclib.pkg
```

## Installing the agent in a VCS environment

Install the agent for WebSphere Application Server on each node in the cluster.

### To install the agent in a VCS environment

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

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

- 2 Uncompress the file to a temporary location, say /tmp.
- 3 If you downloaded the complete Agent Pack tarball, navigate to the directory containing the package for the platform running in your environment.

```
AIX          cd1/aix/vcs/application/websphere_agent/
              vcs_version/version_agent/pkgs

HP-UX        cd1/hpux/generic/vcs/application/websphere_agent/
              vcs_version/version_agent/pkgs

Linux        cd1/linux/generic/vcs/application/websphere_agent/
              vcs_version/version_agent/rpms

Solaris      cd1/solaris/dist_arch/vcs/application/websphere_agent/
              vcs_version/version_agent/pkgs

              where, dist_arch is sol_x64 or sol_sparc
```

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

4 Log in as superuser.

5 Install the package.

```
AIX      # installp -ac -d VRTSvcswas.rte.bff VRTSvcswas.rte
```

```
HP-UX    # swinstall -s 'pwd' VRTSvcswas
```

```
Linux    # rpm -ihv \  
          VRTSvcswas-AgentVersion-GA_GENERIC.noarch.rpm
```

```
Solaris  # pkgadd -d . VRTSvcswas
```

6 After installing the agent package, you must import the agent type configuration file. See [“Importing the agent types files in a VCS environment”](#) on page 24.

## Removing the agent in a VCS environment

You must uninstall the agent for WebSphere Application 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 WebSphere Application Server resources from the cluster. Use the following command to verify that all resources have been removed:

```
# hares -list Type=WebSphere
```

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

```
# hatype -delete WebSphere
```

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 WebSphere Application Server from each node in the cluster.

Execute the following command to uninstall the agent:

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

## Removing the ACC library

Perform the following steps to remove the ACC library.

### To remove the ACC library

- 1 Ensure that all agents that use ACC library are removed.
- 2 Run the following command to remove the ACC library package.

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

# Updating the WebSphere agent to latest release

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.
- # hagrps -freeze group -persistent
- 2
- Stop the cluster services forcibly.
- # hstop -all -force
- 3
- Ensure that the agent operations are stopped on all the nodes.
- # ps -ef |grep WebSphere
- 4
- Uninstall the agent package from all the nodes.
- See “To uninstall the agent in a VCS environment” on page 19.
- 5
- Install the new agent on all the nodes.
- See “To install the agent in a VCS environment” on page 18.
- 6
- Copy the new WebSphereTypes.cf file from the agent's conf directory, to the VCS conf directory /etc/VRTSvcs/conf/config.

VCS Version	Operating system	Agent types files
VCS 4.x	■ AIX	/etc/VRTSvcs/conf/sample_WebSphere/WebSphere.cf
	■ HP-UX	
	■ Linux	
	■ Solaris	
VCS 5.x	■ AIX	/etc/VRTSagents/ha/conf/WebSphere/WebSphereTypes.cf
	■ HP-UX	
	■ Linux	
VCS 5.0	Solaris SPARC and x64	/etc/VRTSagents/ha/conf/WebSphere/WebSphereTypes50.cf
VCS 5.1	Solaris SPARC and x64	/etc/VRTSagents/ha/conf/WebSphere/WebSphereTypes51.cf

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

**8** Start the cluster services.

```
# haagent -start WebSphere -sys System
```

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

```
# hagrps -unfreeze GroupName -persistent
```

# Configuring the agent for WebSphere Application Server

This chapter includes the following topics:

- [About configuring the Veritas agent for WebSphere Application Server](#)
- [Importing the agent types files in a VCS environment](#)
- [WebSphere Application Server agent attributes](#)
- [Uniquely identifying WebSphere Application Server instances](#)
- [Important considerations while configuring the agent](#)
- [Service group configuration options](#)

## About configuring the Veritas agent for WebSphere Application Server

After installing the Veritas agent for WebSphere Application Server, you must import the agent type configuration file. After importing this file, you can create and configure WebSphere Application Server resources. Before you configure a resource, review the attributes table that describes the resource type and its attributes.

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

See [“About sample configurations for the agent for WebSphere Application Server”](#) on page 49.

# Importing the agent types files in a VCS environment

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

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

- 1 Start the Veritas Cluster Manager and connect to the cluster on which the agent is installed.
- 2 Click **File > Import Types**.
- 3 In the Import Types dialog box, select the following file:

VCS version	OperatingSystem	Agent types file
VCS 4.x	■ AIX	/etc/VRTSvcs/conf/sample_WebSphere/
	■ HP-UX	WebSphereTypes.cf
	■ Linux	
	■ Solaris	
VCS 5.x	■ AIX	/etc/VRTSagents/ha/conf/WebSphere/
	■ HP-UX	WebSphereTypes.cf
	■ Linux	
VCS 5.0	■ Solaris SPARC and x64	/etc/VRTSagents/ha/conf/WebSphere/ WebSphereTypes50.cf
VCS 5.1	■ Solaris SPARC and x64	/etc/VRTSagents/ha/conf/WebSphere/ WebSphereTypes51.cf

- 4 Click **Import**.
- 5 Save the VCS configuration.

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

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



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

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

**2** Create a temporary directory.

```
# mkdir ./temp
# cd ./temp
```

**3** Copy the sample file Types.cf.

VCS 4.x	<ul style="list-style-type: none"> <li>■ AIX            /etc/VRTSvcs/conf/sample_WebSphere/</li> <li>■ HP-UX        WebSphereTypes.cf</li> <li>■ Linux</li> <li>■ Solaris</li> </ul>
VCS 5.x	<ul style="list-style-type: none"> <li>■ AIX            /etc/VRTSagents/ha/conf/WebSphere/</li> <li>■ HP-UX        WebSphereTypes.cf</li> <li>■ Linux</li> </ul>
VCS 5.0	<ul style="list-style-type: none"> <li>■ Solaris        /etc/VRTSagents/ha/conf/WebSphere/</li> <li>SPARC            WebSphereTypes50.cf</li> <li>and x64</li> </ul>
VCS 5.1	<ul style="list-style-type: none"> <li>■ Solaris        /etc/VRTSagents/ha/conf/WebSphere/</li> <li>SPARC            WebSphereTypes51.cf</li> <li>and x64</li> </ul>

**4** Create a dummy main.cf file:

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

**5** Create the WebSphere resource type as follows:

```
# hacf -verify .
# haconf -makerw
# sh main.cmd
# haconf -dump
```

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

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

# WebSphere Application Server agent attributes

Table 3-1 shows the required attributes for the agent for WebSphere Application Server.

Table 3-1 Required attributes

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

**Table 3-1** Required attributes (*continued*)

Required attribute	Description
ServerName	<p>Contains the server name assigned to the WebSphere Server during its installation. In Network Deployment configurations, the default ServerName for Deployment Managers is dmgr and the default ServerName for the Node Agents is nodeagent, but these names are not mandatory.</p> <p>See <a href="#">“Uniquely identifying WebSphere Application Server instances”</a> on page 29.</p> <p>Type and dimension: string-scalar</p> <p>Default: ""</p> <p>Example: server1</p>
ServerProfile	<p>Server profile name of the WebSphere Server instance or complete path to the WebSphere Application Server profile.</p> <p>This attribute is applicable to WebSphere version 6.0 and later, and must be null if the WebSphere major version number is 5. You must specify this attribute if the resource manages a WebSphere Application Server version 6.0 and later.</p> <p>Type and dimension: string-scalar</p> <p>Default: ""</p> <p>Example 1: Dmgr01</p> <p>Example 2: /WAS/AppSrv/profiles/AppSrv01/</p>
ServerType	<p>Type of WebSphere Application Server that the cluster will manage. Valid names are as follows:</p> <ul style="list-style-type: none"> <li>■ DeploymentManager: Resource is a Deployment Manager.</li> <li>■ NodeAgent: Resource is a Node Agent.</li> <li>■ ApplicationServer: Resource is an Application Server, which may be a stand-alone server or may be part of a Network Deployment and is a member of a WebSphere Cell.</li> </ul> <p>The agent uses this value to determine how to manage the WebSphere Application Server within a cluster. Refer to the WebSphere documentation for a full explanation of the purposes and use of each WebSphere Application Server type.</p> <p>Type and dimension: string-scalar</p> <p>Default: ""</p> <p>Example: DeploymentManager</p>

**Table 3-1** Required attributes (*continued*)

Required attribute	Description
User	<p>The UNIX user name used to run the programs that start, stop, and monitor the WebSphere resource, which include the program specified in the MonitorProgram attribute. IBM recommends using the root account, but you may use any account. If User is not set to root, the user name must be synchronized across the systems within the cluster. In other words, the user name must resolve to the same UID and have the same default shell on each system in the cluster.</p> <p>Type and dimension: string-scalar</p> <p>Default: ""</p> <p>Example: root</p>
WAS_HOME	<p>The absolute path to the WebSphere Application Server or WebSphere Application Server Network Deployment root installation directory. This attribute is used to locate programs executed by the agent. It is also where the <i>binDir/setupCmdLine.sh</i> file resides. The value is also used to uniquely identify the ServerType processes. Using WAS_HOME to uniquely identify an Application Server's process IDs requires that WAS_HOME be unique compared to WAS_HOME for all other WAS instances in the cluster.</p> <p>See <a href="#">“Uniquely identifying WebSphere Application Server instances”</a> on page 29.</p> <p><b>Note:</b> Both WAS_HOME and WAS_ND_HOME are defined as WAS_HOME in the standard environment file setupCmdLine.sh, which is supplied with WebSphere.</p> <p>Type and dimension: string-scalar</p> <p>Default: ""</p> <p>Example: /ibm/was/v51/cell1/node2</p>
WAS_NODE	<p>The WebSphere Node Name to which the server instance belongs. The Node Name is an administrative identifier that is internal to the WebSphere environment and is assigned when the node is installed. WebSphere requires that a Node Name must be unique within a WebSphere cell.</p> <p>See <a href="#">“Uniquely identifying WebSphere Application Server instances”</a> on page 29.</p> <p>Type and dimension: string-scalar</p> <p>Default: ""</p> <p>Example: was51c1n2</p>

**Table 3-2** lists the optional attributes for the agent for WebSphere Application Server.

**Table 3-2** Optional attributes

Optional Attribute	Definition
MonitorProgram	<p>The full pathname and command-line arguments for an externally-provided custom monitor program. The program is executed within the security context of the UNIX account specified in attribute User. The program must be completely self-contained and independent, and it must return one of the following exit codes:</p> <p>110 or 0: The WebSphere Application Server is ONLINE.</p> <p>100 or 1: The WebSphere Application Server is OFFLINE.</p> <p>All other: The WebSphere Application Server state is UNKNOWN.</p> <p>Symantec recommends storing the external monitor program on the shared storage device, in the directory specified by the WAS_HOME attribute, to ensure the file is always available on the online system.</p> <p>Type and dimension: string-scalar</p> <p>Default: ""</p> <p>Example: /usr/WAS51/server1/bin/mymonitor.sh</p>
StartOptions	<p>The command-line options that are passed to the WebSphere start script when it is executed within the online function. Multiple options should be separated by a space. Refer to the WebSphere product documentation for a list and description of supported start options.</p> <p>Type and dimension: string-scalar</p> <p>Default: ""</p> <p>Example: "-replacelog -trace"</p>
StopOptions	<p>The command-line options that are passed to the WebSphere stop script when it is executed within the offline function. Multiple options should be separated by a space. Refer to the WebSphere product documentation for a list and description of supported stop options.</p> <p>Type and dimension: string-scalar</p> <p>Default: ""</p> <p>Example: "-replacelog -trace"</p>

# Uniquely identifying WebSphere Application Server instances

You can virtualize a WebSphere Application Server instance using a cluster. Using shared disk and virtual IP addresses, you can manage a large set of WebSphere Application Server instances in a single cluster.

Set the WAS\_HOME, WAS\_NODE, and ServerName attributes such that the combined values are unique for each WebSphere Application Server instance.

WebSphere Application Servers can run on separate cluster nodes or can run concurrently on a single node. If WebSphere Application Servers run concurrently on a single node, you must ensure that the agent can uniquely identify each WebSphere Application Server on a host system that is running more than one WebSphere Application Server.

For unique identification, the agent's monitor and clean functions use the values specified by attributes WAS\_HOME, WAS\_NODE, and ServerName to uniquely identify each running WebSphere Server JVM process.

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

## Important considerations while configuring the agent

While configuring the agent, make the following settings:

- The time required to fully start a WebSphere instance depends on the number, size, and complexity of Java applications started within the server. Be sure to compare the value of the OnlineTimeout attribute with the actual time required to fully initialize the WebSphere Application Server. Large WebSphere Application Server deployments may require a larger OnlineTimeout. Properly tuning this attribute ensures that the cluster does not time out the online entry point while a WebSphere Application Server is initializing.
- Allow sufficient time for the WebSphere Application Server to shut down completely before probing the resource to determine if the request to stop was successful. Depending upon the environment, you may need to adjust the OfflineTimeout attribute for this resource to allow the instance ample time to shut down. Properly tuning this attribute ensures that the cluster does not time out the offline entry point while a WebSphere Application Server is completing a graceful shut down.

After a WebSphere Application Server is placed under cluster control, do not attempt to start or stop the instance without using a cluster interface. Only use the Web Console, Java Console, or command-line interface, to start or stop a managed WebSphere instance.

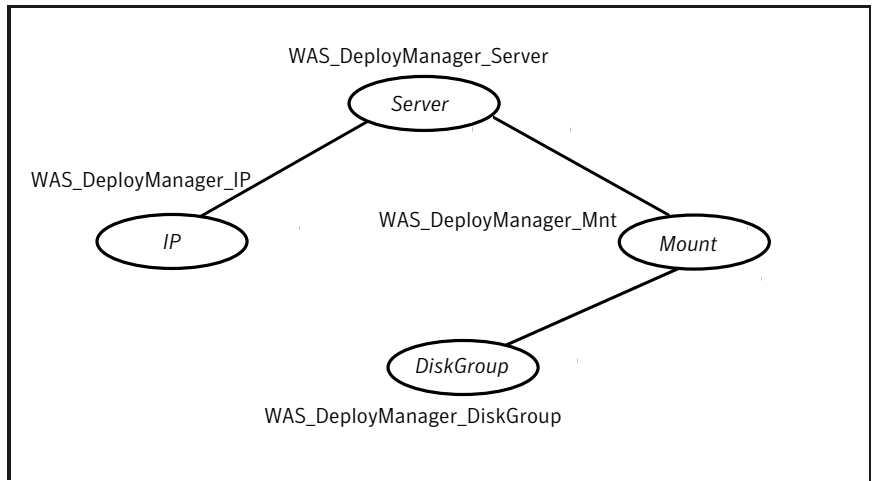
## Service group configuration options

The WebSphere deployment type and strategy determines the number of service groups in a cluster required and the number of WebSphere Application Servers managed within each service group. Although not comprehensive, the following examples depict common scenarios to consider.

[Figure 3-1](#) depicts a service group that manages a Deployment Manager Server.

Other service groups manage WebSphere Servers of the type Node agent and Application Server.

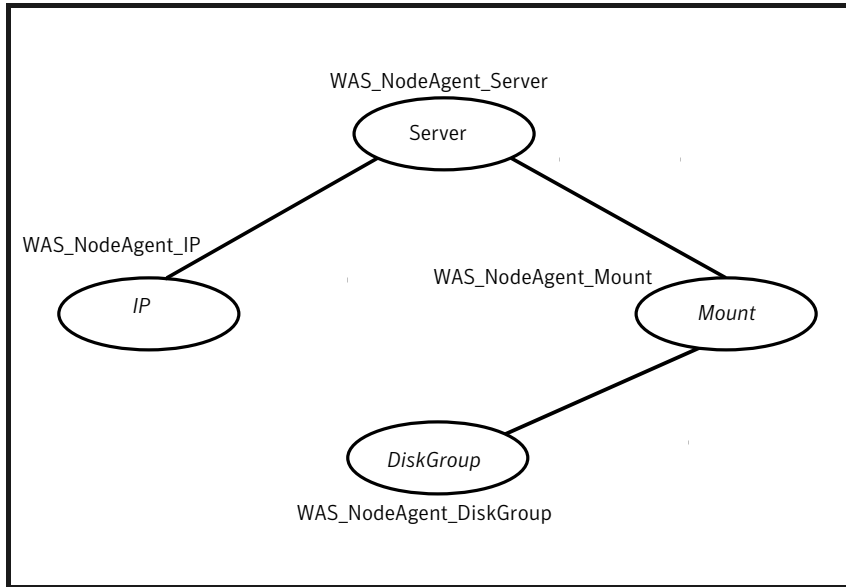
**Figure 3-1** Service group that manages a Deployment Manager Server



[Figure 3-2](#) depicts a service group that manages a Node Agent Server.

In this configuration, the cluster does not control the Application Servers managed by this Node Agent instance. Thus, the Node Agent Server may fully manage and monitor its managed Application Servers without conflict with the cluster.

**Figure 3-2** WebSphere Application Server NodeAgent on all systems

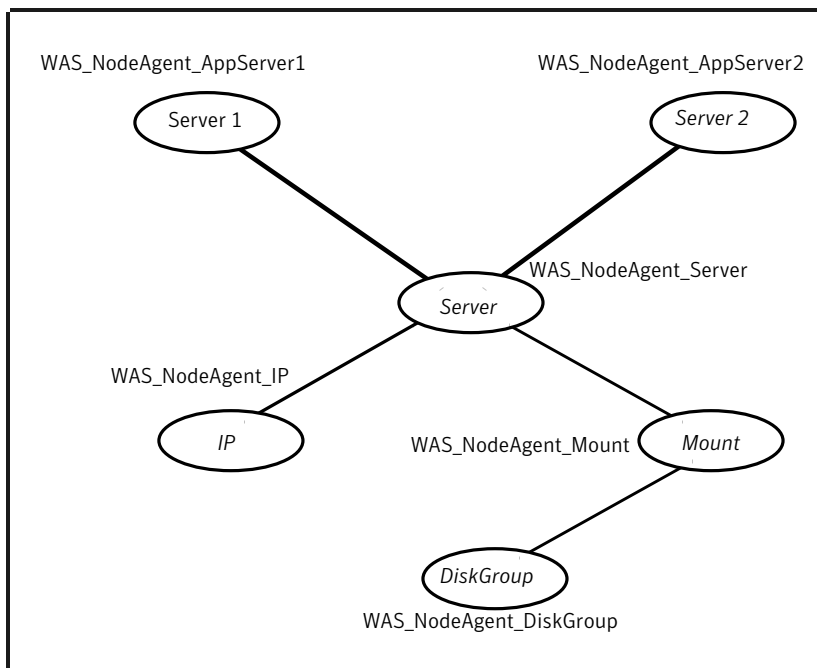


**Figure 3-3** depicts a service group that controls a Node Agent Server and its two managed Application Servers.

In this configuration, the cluster controls the Application Servers that are managed by this Node Agent instance. Thus, the Node Agent Server should be configured to not monitor and restart its failed Application Servers, as this would conflict with cluster actions in response to the failure.



**Figure 3-3** Service group that controls a Node Agent Server and its two managed Application Servers





# Configuring the service groups for WebSphere Application Server

This chapter includes the following topics:

- [Before configuring the service groups for WebSphere Application Server](#)
- [Configuring service groups for WebSphere Application Server](#)

## Before configuring the service groups for WebSphere Application Server

Before you configure the WebSphere Application Server service group, you must:

- Verify that VCS is installed and configured on all nodes in the cluster where you will configure the service group.  
Refer to the *Veritas Cluster Server Installation Guide* for more information.
- Verify that the Veritas agent for WebSphere Application Server is installed on all nodes in the cluster.  
See [“Installing the agent in a VCS environment”](#) on page 18.

# Configuring service groups for WebSphere Application Server

While various methods and procedures can be used to install and cluster a WebSphere Application Server, Symantec recommends the following general process:

## Allocating shared disk resource for the WebSphere node

A WebSphere node is a logical group of WebSphere Application Servers that are located on the same physical machine. This machine is also called a host. Multiple WebSphere nodes can exist on a single node.

Symantec recommends installing each WebSphere node to be clustered on a separate, dedicated shared disk resource (e.g. LUN). Work with the appropriate administrative group in your organization to obtain a shared disk resource for the WebSphere node.

## Creating a Veritas disk group, volume, and file system

Create a Veritas disk group, volume, and file system on the shared disk resource allocated for the WebSphere node.

Although not recommended, WebSphere Application Servers can be clustered without using Veritas Volume Manager or Veritas File System. But the tight integration between the cluster, Volume Manager, and File System ensures a more comprehensive and resilient high availability solution for your WebSphere Application Server.

## Obtaining dedicated virtual IP addresses and host names

Obtain dedicated virtual IP addresses and host names required to support the WebSphere node IP network configuration.

Several configurations are possible. For example, a Node agent, which is an administrative process that manages all servers running on a WebSphere node, can share one IP address and host name with all of its managed servers. Alternatively, the Node agent and each of its managed servers could be assigned its own IP address and host name.

No matter which configuration you deploy, these network addresses and host names will be used exclusively by this WebSphere node, regardless of which system in the cluster is running it.

## Obtaining a dedicated user account if needed

If the WebSphere Application Server will not run using the root account, obtain a dedicated UNIX account for the WebSphere Application Server. Refer to the description of attribute User for important instructions and requirements to create the account.

## Creating service group and supporting resources

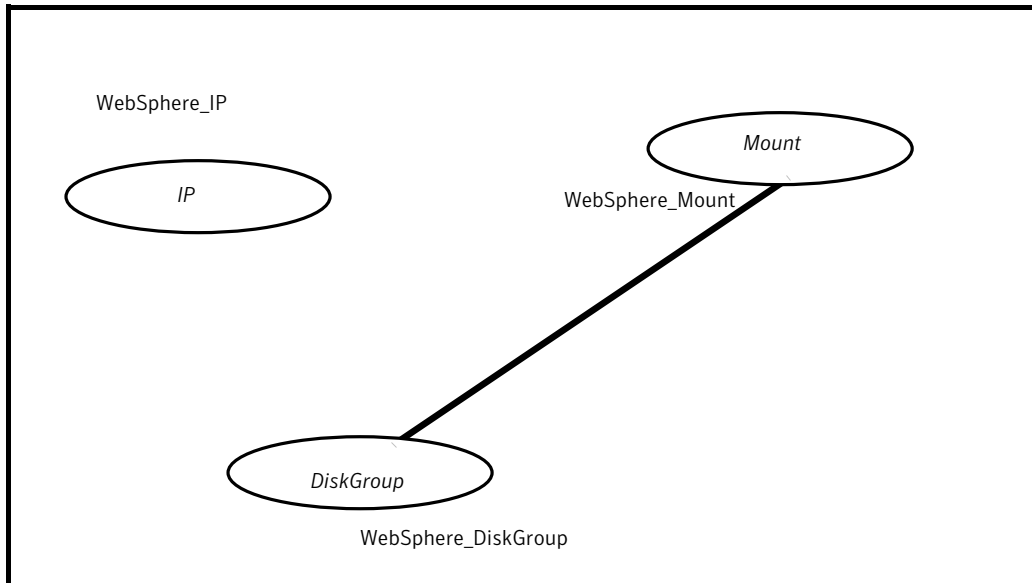
First create a Service Group on a cluster to contain the resources supporting the WebSphere node.

Then create the appropriate cluster resources and links to place the previously created shared disk and networking objects under cluster control.

Test the service group configuration by placing it online. Your service group should appear similar to the following figure.

Figure 4-1 shows a typical service group.

**Figure 4-1** Typical service group



## Installing the WebSphere software

With the disk and network resources now available and online in the cluster, you are ready to install the WebSphere software.

Follow the instructions in the WebSphere product documentation and install the WebSphere Application Server software. Be sure to instruct the installation program to install the software on the shared disk file system previously established for this WebSphere node.

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

Directory	Purpose
/WAS	Top level directory under which all WebSphere nodes are installed.
/WAS/cell1	Subdirectory under which all WebSphere nodes assigned to cell1 are installed.
/WAS/cell1/depmgr	Subdirectory is the mount point for the shared disk resource dedicated to the Deployment Manager instance supporting cell1.
/WAS/cell1/node1	Subdirectory is the mount point for the shared disk resource dedicated to the WebSphere node named <i>node1</i> , which belongs to cell1. The WebSphere software supporting this node agent and its managed Application Servers is installed in this directory.
/WAS/cell1/node2	Subdirectory is the mount point for the shared disk resource dedicated to the WebSphere node named <i>node2</i> , which belongs to cell1. The WebSphere software supporting this node agent and its managed Application Servers is installed in this directory.
/WAS/cell1/node3	Subdirectory is the mount point for the shared disk resource dedicated to the WebSphere node named <i>node3</i> , which belongs to cell1. The WebSphere software supporting this node agent and its managed Application Servers is installed in this directory.

Continue with the same naming pattern for all remaining cells and WebSphere Application Servers.

During the installation, be sure to set the node's Host Name to the dedicated virtual IP host name previously allocated to this node.

Finally, be sure to configure the server's port numbers to avoid conflicts with the port numbers of other WebSphere Application Servers that may be running

simultaneously on the same system. Configuring the port numbers is especially important in a cluster environment where WebSphere nodes can be easily moved around the systems in the cluster in almost any combination.

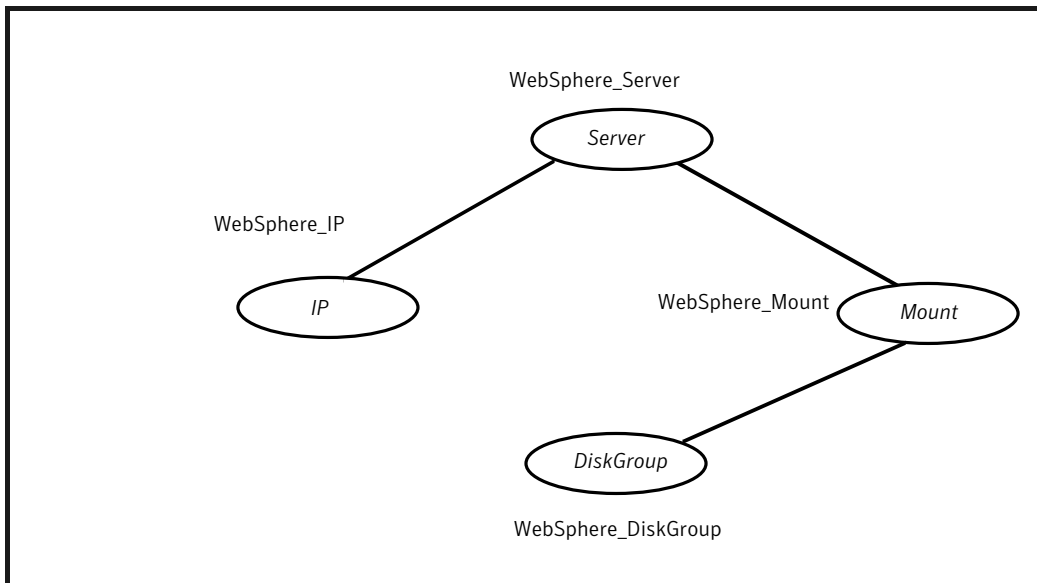
## Placing the WebSphere Application Server under cluster control

After the WebSphere Application Server installation is complete, create a cluster resource using the agent for WebSphere to place the server under cluster control.

Your service group should now appear similar to the following figure.

Figure 4-2 shows a typical service group.

**Figure 4-2** Typical service group



---

**Warning:** After a WebSphere Application Server is placed under cluster control, do not attempt to start or stop the instance without using a cluster interface. Only use the Web Console, Java Console, or command-line interface to start or stop a managed WebSphere instance.

---





# Troubleshooting the agent for WebSphere Application Server

This chapter includes the following topics:

- [Using the correct software and operating system versions](#)
- [Meeting prerequisites](#)
- [Configuring WebSphere Application Server resources](#)
- [Starting the WebSphere Application Server instance outside a cluster](#)
- [Reviewing error log files](#)
- [Defining additional environment variables for a WebSphere Application Server instance](#)

## Using the correct software and operating system versions

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

See [“Supported software”](#) on page 10.

## Meeting prerequisites

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

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

See [“Before you install the Veritas agent for WebSphere Application Server”](#) on page 15.

## Configuring WebSphere Application Server resources

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

## Starting the WebSphere Application 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 WebSphere Application Server instance independent of the cluster framework. Refer to the cluster documentation for information about disabling a resource.

You can then restart the WebSphere Application 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 WebSphere instance outside the cluster framework, is illustrated as follows.

### To start a WebSphere Deployment Manager outside the cluster framework

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

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

Attribute	Value
ServerType	DeploymentManager
ServerName	dmgr
WAS_NODE	was60c1dmsol
WAS_HOME	/ibm/was/v60/cell1/depmgr
ServerProfile	Dmgr01Specify this attribute for WebSphere version 6.0 and later

- 3 Go to specified directory.

5.x                    /ibm/was/v60/cell1/depmgr/bin

6.x                    /ibm/was/v60/cell1/depmgr/profiles/Dmgr0/bin

- 4 Using the startManager.sh script, start the Deployment Manager.

5.x                    /ibm/was/v60/cell1/depmgr/bin/startManager.sh

6.x                    /ibm/was/v60/cell1/depmgr/profiles/Dmgr01/bin/startManager.sh

- 5 Ensure that the Deployment Manager Server starts successfully.

If the Deployment Manager works properly outside the cluster framework, you can attempt to implement the server within the framework.

**To start a WebSphere node agent outside the cluster framework**

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

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

Attribute	Value
ServerType	NodeAgent
ServerName	nodeagent
WAS_NODE	was60c1n1sol
WAS_HOME	/ibm/was/v60/cell1/node1
ServerProfile	Default

- 3
- Go to specified directory.

WebSphere version	Directory
5.x	/ibm/was/v60/cell1/node1/bin
6.x	/ibm/was/v60/cell1/node1/profiles/default/bin

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

5.x	/ibm/was/v60/cell1/node1/bin/startNode.sh
6.x	/ibm/was/v60/cell1/node1/profiles/default/bin/startNode.sh

- 5
- Ensure that the Node Agent starts successfully.

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

# Reviewing error log files

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

## Reviewing cluster log files

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

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

## Reviewing agent log files

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

*/tmp/.Resource\_Name.Entry\_Point.Process\_ID*

For example, for a resource `WAS50DeployMgr_dmgr`:

*/tmp/.WAS50DeployMgr\_dmgr.online.Process\_ID*  
*/tmp/.WAS50DeployMgr\_dmgr.offline.Process\_ID*  
*/tmp/.WAS50DeployMgr\_dmgr.clean.Process\_ID*  
*/tmp/.WAS50DeployMgr\_dmgr.monitor.Process\_ID*

If a resource, `WAS50DeployMgr_dmgr` is unable to bring a WebSphere Node Manager online, you can access the */tmp/.WAS50DeployMgr\_dmgr.online.Process\_ID* for more information so that you can diagnose the problem.

---

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

---

## Using trace level logging

The `ResLogLevel` attribute controls the level of logging that is written in a cluster log file for each WebSphere Application Server resource. You can set this attribute to `TRACE`, which enables very detailed and verbose logging.

If you set ResLogLevel to TRACE, a very high volume of messages are produced. Symantec recommends that you localize the ResLogLevel attribute for a particular 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.

## Defining additional environment variables for a WebSphere Application Server instance

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

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

#!/bin/sh

binDir=`dirname ${0}`

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

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

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

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

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

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





# Sample Configurations

This appendix includes the following topics:

- [About sample configurations for the agent for WebSphere Application Server](#)
- [Sample agent type definition for WebSphere Application Server](#)
- [Sample configuration in a VCS environment](#)

## About sample configurations for the agent for WebSphere Application 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 WebSphere Application Server. For more information about these resource types, see the *Veritas Cluster Server Bundled Agents Reference Guide*.

## Sample agent type definition for WebSphere Application Server

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 `WebSphereTypes.cf` file in the `/etc/VRTSvcs/conf/config` cluster configuration directory.

An excerpt from this file follows.

```
type WebSphere (
```

```
static str ArgList [] = { ResLogLevel, State, IState, ServerName,
    WAS_NODE, WAS_HOME ,User, ServerProfile, ServerType,
    StartOptions, StopOptions, MonitorProgram, SecondLevelMonitor}
str ResLogLevel = INFO
str ServerName
str WAS_NODE
str WAS_HOME
str User
str ServerProfile
str ServerType
str StartOptions
str StopOptions
str MonitorProgram
int SecondLevelMonitor
)
```

### VCS 5.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 `WebSphereTypes.cf` file in the `/etc/VRTSagents/ha/conf/config` cluster configuration directory.

An excerpt from this file follows.

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

## Sample configuration in a VCS environment

The following is an excerpt from a VCS configuration file (main.cf) that defines a Network Deployment of WebSphere Application Servers and two independent Application Servers (Application Servers that are not part of a Network Deployment).

This configuration demonstrates that you can combine Network Deployment WebSphere Cells with independent WebSphere Application Servers. In the example, there is one WebSphere Cell consisting of one Deployment Manager named `dmgr`. The WebSphere Cell contains two Node Managers, both named `nodeagent`.

Review the information to configure a service group that manages one independent, stand-alone Application Server.

See [“Service group configuration options”](#) on page 31.

```
group WAS51Cell1DM (
    SystemList = { sysa = 0, sysb = 1, sysc = 2 }
)
DiskGroup WAS51Cell1DM_dg (
    DiskGroup = was51c1dm
)

IP WAS51Cell1DM_ip (
    Device = hme0
    Address = "10.136.228.11"
    NetMask = "255.255.248.0"
)

Mount WAS51Cell1DM_mnt (
    MountPoint = "/WAS51/cell1/depmgr"
    BlockDevice = "/dev/vx/dsk/was51c1dm/was"
    FSType = vxfs
    FsckOpt = "-y"
)

WebSphere WAS51Cell1DM_was (
    ServerName = dmgr
    WAS_NODE = was51c1dm
    WAS_HOME = "/WAS51/cell1/depmgr"
    User = root
    ServerType = DeploymentManager
)
```

```

    )

    WAS51Cell11DM_was requires WAS51Cell11DM_ip
    WAS51Cell11DM_was requires WAS51Cell11DM_mnt
    WAS51Cell11DM_mnt requires WAS51Cell11DM_dg

group WAS51Cell11Node1 (
    SystemList = { sysb = 0, sysa = 1, sysc = 2 }
)
DiskGroup WAS51Cell11Node1_dg (
    DiskGroup = was51c1n1
)

IP WAS51Cell11Node1_ip (
    Device = hme0
    Address = "10.136.228.18"
    NetMask = "255.255.248.0"
)

Mount WAS51Cell11Node1_mnt (
    MountPoint = "/WAS51/cell11/node1"
    BlockDevice = "/dev/vx/dsk/was51c1n1/was"
    FSType = vxfs
    FsckOpt = "-y"
)

WebSphere WAS51Cell11Node1_NA_was (
    ServerName = nodeagent
    WAS_NODE = was51c1n1
    WAS_HOME = "/WAS51/cell11/node1"
    User = root
    ServerType = NodeAgent
)

WebSphere WAS51Cell11Node1_AS1_was (
    ServerName = server1
    WAS_NODE = was51c1n1
    WAS_HOME = "/WAS51/cell11/node1"
    User = root
    ServerType = ApplicationServer
)

```

```
WebSphere WAS51Cell11Node1_AS2_was (
    ServerName = server2
    WAS_NODE = was51c1n1
    WAS_HOME = "/WAS51/cell11/node1"
    User = root
    ServerType = ApplicationServer
)

WAS51Cell11Node1_AS2_was requires WAS51Cell11Node1_NA_was
WAS51Cell11Node1_mnt requires WAS51Cell11Node1_dg
WAS51Cell11Node1_NA_was requires WAS51Cell11Node1_ip
WAS51Cell11Node1_NA_was requires WAS51Cell11Node1_mnt
WAS51Cell11Node1_AS1_was requires WAS51Cell11Node1_NA_was

group WAS51Cell11Node2 (
    SystemList = { sysa = 0, sysa = 1, sysb = 2 }
)

DiskGroup WAS51Cell11Node2_dg (
    DiskGroup = was51c1n2
)

IP WAS51Cell11Node2_ip (
    Device = eri0
    Address = "10.136.228.19"
    NetMask = "255.255.248.0"
)

Mount WAS51Cell11Node2_mnt (
    MountPoint = "/WAS51/cell11/node2"
    BlockDevice = "/dev/vx/dsk/was51c1n2/was"
    FSType = vxfs
    FsckOpt = "-y"
)

WebSphere WAS51Cell11Node2_NA_was (
    ServerName = nodeagent
    WAS_NODE = was51c1n2
    WAS_HOME = "/WAS51/cell11/node2"
    User = root
    ServerType = NodeAgent
)
```

```
WebSphere WAS51Cell11Node2_AS1_was (
    ServerName = server1
    WAS_NODE = was51c1n2
    WAS_HOME = "/WAS51/cell11/node2"
    User = root
    ServerType = ApplicationServer
)

WAS51Cell11Node2_mnt requires WAS51Cell11Node2_dg
WAS51Cell11Node2_NA_was requires WAS51Cell11Node2_ip
WAS51Cell11Node2_NA_was requires WAS51Cell11Node2_mnt
WAS51Cell11Node2_AS1_was requires WAS51Cell11Node2_NA_was

group WAS51AppSrvr1 (
    SystemList = { sysa = 0, sysb = 1, sysc = 2 }
)

DiskGroup WAS51AppSrvr1_dg (
    DiskGroup = was51appsrvr1
)

IP WAS51AppSrvr1_ip (
    Device = hme0
    Address = "10.136.228.21"
    NetMask = "255.255.248.0"
)

Mount WAS51AppSrvr1_mnt (
    MountPoint = "/WAS51/srvr1"
    BlockDevice = "/dev/vx/dsk/was51appsrvr1/was"
    FSType = vxfs
    FsckOpt = "-y"
)

WebSphere WAS51AppSrvr1_AppSrvr_was (
    ServerName = server1
    WAS_NODE = was51srv1
    WAS_HOME = "/WAS51/srvr1"
    User = root
    ServerType = ApplicationServer
)
```

```
WAS51AppSrvr1_mnt requires WAS51AppSrvr1_dg
WAS51AppSrvr1_AppSrvr_was requires WAS51AppSrvr1_ip
WAS51AppSrvr1_AppSrvr_was requires WAS51AppSrvr1_mnt

group WAS51AppSrvr2 (
    SystemList = { sysa = 0, sysb = 1, sysc = 2 }
)

DiskGroup WAS51AppSrvr2_dg (
    DiskGroup = was51appsrvr2
)

IP WAS51AppSrvr2_ip (
    Device = hme0
    Address = "10.136.228.22"
    NetMask = "255.255.248.0"
)

Mount WAS51AppSrvr2_mnt (
    MountPoint = "/WAS51/srvr2"
    BlockDevice = "/dev/vx/dsk/was51appsrvr2/was"
    FSType = vxfs
    FsckOpt = "-y"
)

WebSphere WAS51AppSrvr2_AppSrvr_was (
    ServerName = server1
    WAS_NODE = was51srv1
    WAS_HOME = "/WAS51/srvr2"
    User = root
    ServerType = ApplicationServer
)

WAS51AppSrvr2_mnt requires WAS51AppSrvr2_dg
WAS51AppSrvr2_AppSrvr_was requires WAS51AppSrvr2_ip
WAS51AppSrvr2_AppSrvr_was requires WAS51AppSrvr2_mnt
```





# Changes introduced in previous releases

This appendix includes the following topics:

- [Changes introduced in previous releases](#)

## Changes introduced in previous releases

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

- Fixed issue that arose due to incorrect timeout value. The WebSphere agent was using incorrect timeout values in certain cases to set the alarm for execution of the commands. Now the agent is using the correct timeout value for the entry point operations.
- Added support for IPv6.
- Added support for WebSphere Application Server 7.0.
- Added support for AIX 6.1 on pSeries.
- Added support for VCS 5.1 on AIX, Linux, and Solaris.
- Added support for SUSE Linux Enterprise Server 9.
- Added support for VCS One 2.0 and 5.0.
- Added support for Solaris x86 for VCS 4.1 and 5.0.
- Added support for Internationalization (i18n).
- Added support for VCS version 5.0.
- Added support for ACC library 5.1 that is compliant with VCS 4.x.

- Added support for zones on Solaris 10 in a VCS environment.
- Renamed agent type WebSphere5 to agent type WebSphere.

# Index

## A

- about ACC library 17
- ACC library
  - installing 17
  - removing 20
- agent
  - configuring service groups 36
  - i18n support 16
  - importing agent types files 24
  - installing, VCS environment 18
  - overview 9
  - supported software 10
  - uninstalling, VCS environment 19
  - what's new 10
- agent attributes
  - MonitorProgram 29
  - ResLogLevel 26
  - SecondLevelMonitor 26
  - ServerName 27
  - ServerProfile 27
  - ServerType 27
  - StartOptions 29
  - StopOptions 29
  - User 28
  - WAS\_HOME 28
  - WAS\_NODE 28
- agent configuration file
  - importing 24
- agent functions
  - clean 13
  - monitor 12
- agent installation
  - general requirements 15
  - steps to install 18

## B

- before
  - configuring the service groups 35

## L

- logs
  - reviewing cluster log files 45
  - reviewing error log files 45
  - using trace level logging 45

## R

- removing agent, VCS environment 19

## S

- sample configurations
  - VCS environment 51
- starting the WebSphere Application Server instance
  - outside a cluster 42
- supported software 10

## T

- troubleshooting
  - meeting prerequisites 42
  - reviewing error log files 45
    - reviewing cluster log files 45
  - using trace level logging 45
  - using correct software 41

## U

- uninstalling agent, VCS environment 19
- uniquely identifying WebSphere Application server instances 30

## W

- WebSphere Application Server
  - configuring resources 42
  - starting instance outside cluster 42