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

AIX, HP-UX, Linux, Solaris

5.1

December 2008

symantec™

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

Agent version: 5.1

Document version: 5.1.2

Legal Notice

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 WebSphere Application Server

This chapter includes the following topics:

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

What's new in this agent

The enhancements in this release of WebSphere 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 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.

Supported software

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

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
Veritas Cluster Server One	VCS One 2.0 on AIX, HP-UX, Linux, and Solaris
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	■ 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 in both VCS and VCS One environments.
WebSphere Application Server	5.0, 5.1, 6.0, 6.1

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	`WAS_HOME/binDir/startManager.sh`
Node Agent	`WAS_HOME/binDir/startNode.sh`
Application Server	`WAS_HOME/binDir/startServer.sh`

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	`WAS_HOME/binDir/stopManager.sh`
Node Agent	`WAS_HOME/binDir/stopNode.sh`
Application Server	`WAS_HOME/binDir/stopServer.sh`

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 or VCS One administrators 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 28.

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.

Chapter 2

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

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

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.
- Remove any previous version of this agent.
- 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 "About ACC Library" on page 17.

For VCS One, do the following:

- Install and configure Veritas Cluster Server One.
- Remove any previous version of this agent.

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 WebSphere Application 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 WebSphere Application 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 WebSphere Application 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 WebSphere Application 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/websphere_agent/vcs_version/version_agent/pkgs`

   HP-UX `cd_mount/hpux/generic/application/websphere_agent/vcs_version/version_agent/pkgs`

   Linux `cd_mount/linux/generic/application/websphere_agent/vcs_version/version_agent/rpms`

   Solaris `cd_mount/solaris/dist_arch/application/websphere_agent/vcs_version/version_agent/pkgs`

3. Install the package.

   AIX
   ```
   # installp -ac -d VRTSvcswas5.rte.bff VRTSvcswas5.rte
   ```

   HP-UX
   ```
   # swinstall -s 'pwd' VRTSvcswas5
   ```

   Linux
   ```
   # rpm -ihv \
   VRTSvcswas5-AgentVersion-GA_GENERIC.noarch.rpm
   ```

   Solaris
   ```
   # pkgadd -d . VRTSvcswas5
   ```

Installing the agent in a VCS One environment

You must install the agent for WebSphere Application Server on all the application nodes of the server farm that will host the WebSphere service group.

You can install the agent for WebSphere Application Server using the installagpack program.

The installagpack program installs the following agents:

- Oracle

- Sybase
- WebLogic Server
- WebSphere Application Server
- WebSphere MQ

  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 for VCS One" on page 26.
-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 high availability agents

Installing the agent packs involves the following phases:

- Installing the agent packages
- Adding the agent resource type definitions

  See "Adding the agent type definitions for VCS One" on page 26.

Installing the agent packages

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

To install the Veritas high availability agents

1. Mount the Agent Pack software disc on the application node 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 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_arch* is sparc or sol_x64

3. Enter the command to start the agent pack installation:

```
# ./installagpack
```

4. Enter the name of an application node or application nodes where you want to install the agents.
5. 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 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=WebSphere5
```

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

```
# hatype -delete WebSphere5
```

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 VRTSvcswas5.rte`
HP-UX	`# swremove VRTSvcswas5`
Linux	`# rpm -e VRTSvcswas5`
Solaris	`# pkgrm VRTSvcswas5`

Removing the agent in a VCS One environment

You can remove the Veritas high availability agents that you installed using the installagpack program.

To remove the Veritas high availability agents from application nodes

1. Mount the Agent Pack software disc on the application node 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 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_arch* is sparc or sol_x64

3. Start the uninstallagpack program.

```
# ./uninstallagpack
```

4. Enter the name of the application nodes separated by spaces on which you want to uninstall the agent pack.
5. 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 for WebSphere Application Server

To upgrade the agent, first remove the older version of the agent.

See "Removing the agent in a VCS environment" on page 21.

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

Then, follow the instructions to install the new agent software.

See "Installing the agent in a VCS environment" on page 18.

See "Installing the agent in a VCS One environment" on page 19.

Chapter 3

Preparing to configure the agent for WebSphere Application Server

This chapter includes the following topics:

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 a WebSphere Application 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 WebSphere Application Server" on page 49.

Importing the agent types files for VCS

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 4.x	/etc/VRTSvcs/conf/sample_WebSphere5/WebSphere5Types.cf
VCS 5.0	/etc/VRTSagents/ha/conf/WebSphere5/WebSphere5Types.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*.

Adding the agent type definitions for VCS One

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 application node 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 application node where you ran the installer, then you must set up rsh or paswordless ssh communications between this application node 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 application node 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_arch* is sparc or sol_x64

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 an application node, 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.

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	The logging detail performed by the agent 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 functions. Type and dimension: string-scalar Default: INFO Example: TRACE

Table 3-1 Required attributes *(continued)*

Required attribute	Description
SecondLevelMonitor	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. 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. Type and dimension: integer-scalar Default: 0 Example: 1
ServerName	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. See "Uniquely identifying WebSphere Application Server instances" on page 32. Type and dimension: string-scalar Default: "" Example: server1
ServerProfile	Server profile name of the WebSphere Server instance or complete path to the WebSphere Application Server profile. 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. Type and dimension: string-scalar Default: "" Example 1: Dmgr01 Example 2: /WAS/AppSrv/profiles/AppSrv01/

Table 3-1 Required attributes *(continued)*

Required attribute	Description
ServerType	Type of WebSphere Application Server that the cluster will manage. Valid names are as follows: ■ DeploymentManager: Resource is a Deployment Manager. ■ NodeAgent: Resource is a Node Agent. ■ 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. 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. Type and dimension: string-scalar Default: "" Example: DeploymentManager
User	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. Type and dimension: string-scalar Default: "" Example: root
WAS_HOME	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 *binDir*/setupCmdLine.sh 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. See "Uniquely identifying WebSphere Application Server instances" on page 32. **Note:** Both WAS_HOME and WAS_ND_HOME are defined as WAS_HOME in the standard environment file setupCmdLine.sh, which is supplied with WebSphere. Type and dimension: string-scalar Default: "" Example: /ibm/was/v51/cell1/node2

Table 3-1 Required attributes *(continued)*

Required attribute	Description
WAS_NODE	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. See "Uniquely identifying WebSphere Application Server instances" on page 32. Type and dimension: string-scalar Default: "" Example: was51c1n2

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

Table 3-2 Optional attributes

Optional Attribute	Definition
MonitorProgram	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: 110 or 0: The WebSphere Application Server is ONLINE. 100 or 1: The WebSphere Application Server is OFFLINE. All other: The WebSphere Application Server state is UNKNOWN. 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. Type and dimension: string-scalar Default: "" Example: /usr/WAS51/server1/bin/mymonitor.sh
StartOptions	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. Type and dimension: string-scalar Default: "" Example: "-replacelog -trace"

Table 3-2 Optional attributes *(continued)*

Optional Attribute	Definition
StopOptions	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. Type and dimension: string-scalar Default: "" Example: "-replacelog -trace"

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

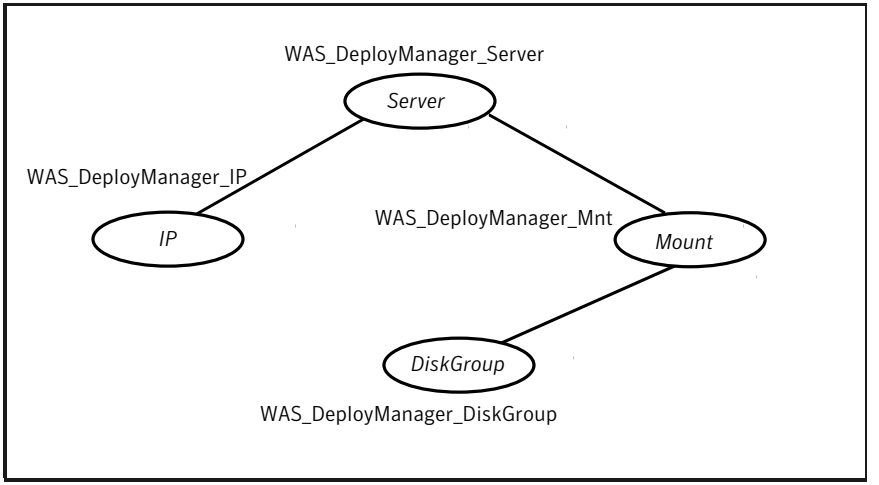


Figure 3-2 depicts a service group that manages a Node Agent Server.

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

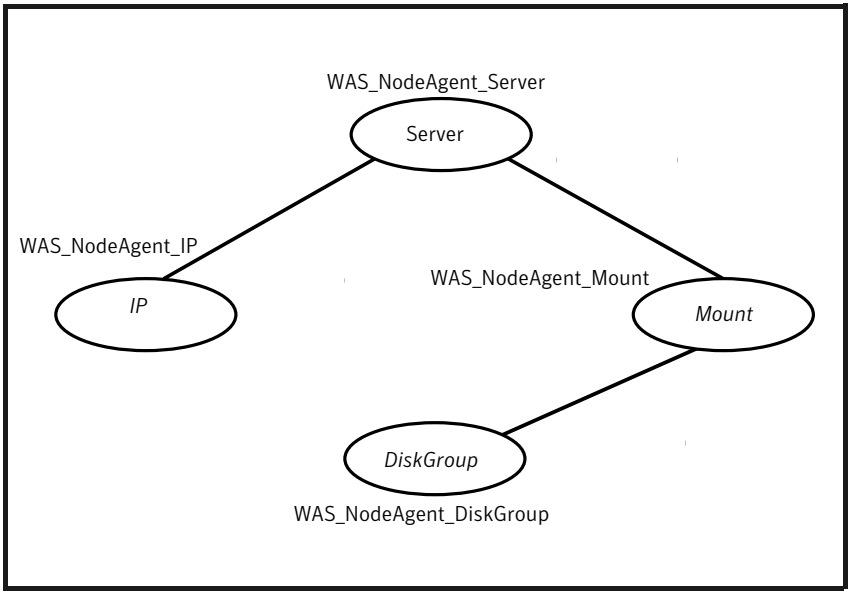
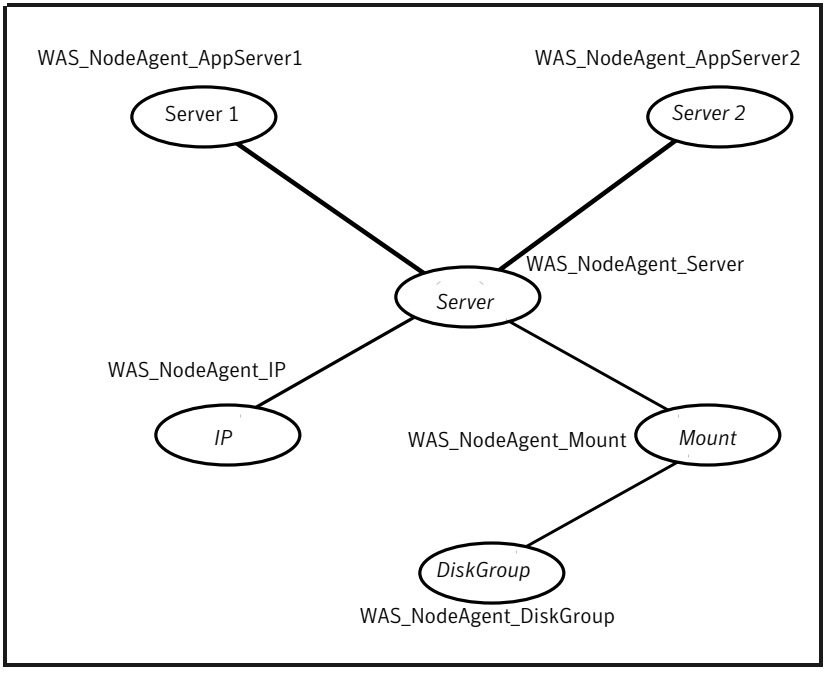


Figure 3-3 depicts a service group that controls a Node Agent Server and its two managed Application Servers.

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



Chapter 4

Configuring the service groups for WebSphere Application Server

This chapter includes the following topics:

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

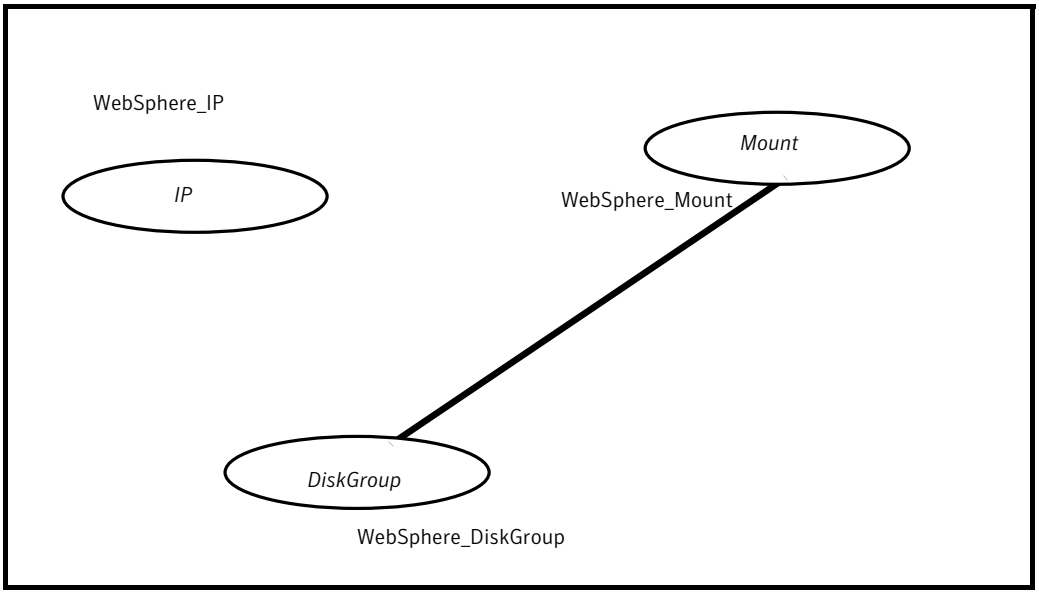
First create a Service Group on a cluster to contain the resources supporting the WebSphere node.

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 *node1*, 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 node2, 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 node3, 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.

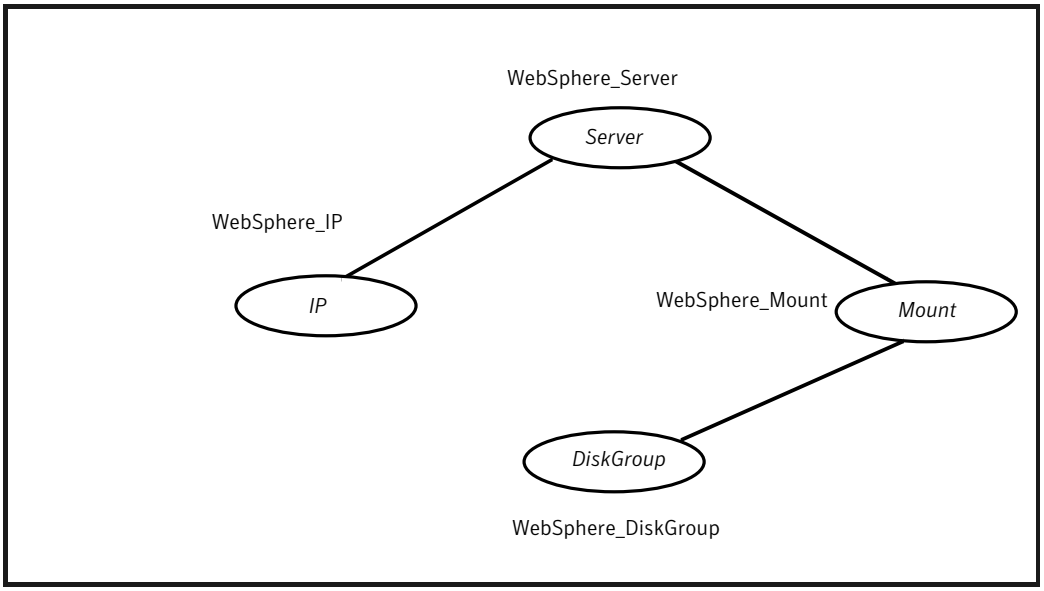
Placing the WebSphere Application Server under cluster control

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.

Chapter 5

Troubleshooting the agent for WebSphere Application 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 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 a WebSphere Application Server resource, ensure that you configure the resource 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
6.0	/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`
6.0	`/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
6.0	/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`
6.0	`/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.
- 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.

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.

Appendix A

Sample Configurations

This appendix includes the following topics:

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 WebSphere5Types.cf file in the /etc/VRTSvcs/conf/config cluster configuration directory.

An excerpt from this file follows.

```
type WebSphere5 (
   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 5.0

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 WebSphere5Types.cf file in the /etc/VRTSagents/ha/conf/config cluster configuration directory.

An excerpt from this file follows.

```
type WebSphere5 (
    static str AgentFile = "/opt/VRTSvcs/bin/Script50Agent"
    static str AgentDirectory = "/opt/VRTSagents/ha/bin/WebSphere5"
   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 33.

```
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"
                )

        WebSphere5 WAS51Cell1DM_was (
                ServerName = dmgr
```

```
                WAS_NODE = was51c1dm
                WAS_HOME = "/WAS51/cell1/depmgr"
                User = root
                ServerType = DeploymentManager
                )

        WAS51Cell1DM_was requires WAS51Cell1DM_ip
        WAS51Cell1DM_was requires WAS51Cell1DM_mnt
        WAS51Cell1DM_mnt requires WAS51Cell1DM_dg


group WAS51Cell1Node1 (
        SystemList = { sysb = 0, sysa = 1, sysc = 2 }
        )
        DiskGroup WAS51Cell1Node1_dg (
                DiskGroup = was51c1n1
                )

        IP WAS51Cell1Node1_ip (
                Device = hme0
                Address = "10.136.228.18"
                NetMask = "255.255.248.0"
                )

        Mount WAS51Cell1Node1_mnt (
                MountPoint = "/WAS51/cell1/node1"
                BlockDevice = "/dev/vx/dsk/was51c1n1/was"
                FSType = vxfs
                FsckOpt = "-y"
                )

        WebSphere5 WAS51Cell1Node1_NA_was (
                ServerName = nodeagent
                WAS_NODE = was51c1n1
                WAS_HOME = "/WAS51/cell1/node1"
                User = root
                ServerType = NodeAgent
                )

        WebSphere5 WAS51Cell1Node1_AS1_was (
                ServerName = server1
                WAS_NODE = was51c1n1
                WAS_HOME = "/WAS51/cell1/node1"
```

```
                User = root
                ServerType = ApplicationServer
                )

        WebSphere5 WAS51Cell1Node1_AS2_was (
                ServerName = server2
                WAS_NODE = was51c1n1
                WAS_HOME = "/WAS51/cell1/node1"
                User = root
                ServerType = ApplicationServer
                )

        WAS51Cell1Node1_AS2_was requires WAS51Cell1Node1_NA_was
        WAS51Cell1Node1_mnt requires WAS51Cell1Node1_dg
        WAS51Cell1Node1_NA_was requires WAS51Cell1Node1_ip
        WAS51Cell1Node1_NA_was requires WAS51Cell1Node1_mnt
        WAS51Cell1Node1_AS1_was requires WAS51Cell1Node1_NA_was


group WAS51Cell1Node2 (
        SystemList = { sysa = 0, sysa = 1, sysb = 2 }
        )

        DiskGroup WAS51Cell1Node2_dg (
                DiskGroup = was51c1n2
                )

        IP WAS51Cell1Node2_ip (
                Device = eri0
                Address = "10.136.228.19"
                NetMask = "255.255.248.0"
                )

        Mount WAS51Cell1Node2_mnt (
                MountPoint = "/WAS51/cell1/node2"
                BlockDevice = "/dev/vx/dsk/was51c1n2/was"
                FSType = vxfs
                FsckOpt = "-y"
                )

        WebSphere5 WAS51Cell1Node2_NA_was (
                ServerName = nodeagent
                WAS_NODE = was51c1n2
```

```
                WAS_HOME = "/WAS51/cell1/node2"
                User = root
                ServerType = NodeAgent
                )

        WebSphere5 WAS51Cell1Node2_AS1_was (
                ServerName = server1
                WAS_NODE = was51c1n2
                WAS_HOME = "/WAS51/cell1/node2"
                User = root
                ServerType = ApplicationServer
                )

        WAS51Cell1Node2_mnt requires WAS51Cell1Node2_dg
        WAS51Cell1Node2_NA_was requires WAS51Cell1Node2_ip
        WAS51Cell1Node2_NA_was requires WAS51Cell1Node2_mnt
        WAS51Cell1Node2_AS1_was requires WAS51Cell1Node2_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"
                )

        WebSphere5 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"
                )

        WebSphere5 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
```

Sample configuration in a VCS One environment

A sample main.xml file is in the /etc/VRTSagents/ha/conf/WebSphere5/ directory.

This sample configuration defines a Network Deployment of WebSphere Application Servers and an independent Application Server, which is not part of a Network Deployment.

Index