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

Windows

6.1

June 2015

Symantec

Symantec 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: 6.1

Document version: 6.1 Rev 1

Legal Notice

Copyright © 2015 Symantec Corporation. All rights reserved.

Symantec, the Symantec Logo, the Checkmark Logo, Veritas, the Veritas Logo, 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 website at the following URL:

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

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

Contacting Technical Support

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

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

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

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

- Product release level
- Hardware information

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

Licensing and registration

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

www.symantec.com/business/support/

Customer service

Customer service information is available at the following URL:

www.symantec.com/business/support/

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

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

Support agreement resources

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

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

Documentation

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

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

For information regarding the latest HOWTO articles, documentation updates, or to ask a question regarding product documentation, visit the Storage and Clustering Documentation forum on Symantec Connect.

https://www-secure.symantec.com/connect/storage-management/forums/storage-and-clustering-documentation

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 3

Chapter 1 Introducing the Symantec High Availability Agent for WebSphere Application Server 8

About the Symantec High Availability agent for WebSphere Application Server 8
Supported software 9
Agent functions 9
Online 9
Offline 9
Monitor 10
Clean 11

Chapter 2 Installing, upgrading, and removing the agent for WebSphere Application Server 12

Before you install the agent for WebSphere Application Server 12
Installing the VCS agent for WebSphere Application Server 12
Removing the VCS agent for WebSphere Application Server 13

Chapter 3 Configuring the agent for WebSphere Application Server 15

About configuring the agent for WebSphere Application Server 15
WebSphere Application Server agent attributes 15
Uniquely identifying WebSphere Application Server instances 20
Executing a custom monitor program 20
Service group configuration options 21
Important considerations while configuring the agent 23

Chapter 4 Configuring the service groups for WebSphere Application Server 25

Configuring service groups for WebSphere Application Server 25
Allocating shared disk resource for the WebSphere node 25
Creating a Veritas disk group, volume, and file system 26

Obtaining dedicated virtual IP addresses and host names 26
Obtaining a dedicated user account if needed 26
Creating service group and supporting resources 26
Installing the WebSphere software 27
Placing the WebSphere Application Server under cluster control 28

Chapter 5 Troubleshooting the agent for WebSphere Application Server 30

Using correct software and operating system versions 30
Starting the WebSphere Application Server outside a cluster 30
Reviewing error log files 32
Reviewing VCS log files 32
Reviewing agent log files 33
Using trace level logging 33

Appendix A Sample Configurations 34

About the sample configuration for the agent for WebSphere Application Server 34
Sample WebSphere Application Server agent type definition 34
Sample configuration 35
Sample service group configuration for the agent for WebSphere Application Server 42

Index 44

Chapter 1

Introducing the Symantec High Availability Agent for WebSphere Application Server

This chapter includes the following topics:

- About the Symantec High Availability agent for WebSphere Application Server
- Supported software
- Agent functions

About the Symantec High Availability agent for WebSphere Application Server

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

The Symantec High Availability agent for WebSphere Application Server monitors specific resources within an enterprise application, determines the status of these resources, and starts or stops them according to external events. The agent for WebSphere Application Server provides high availability for WebSphere Application Server in a cluster environment.

High availability is supported for following types of WebSphere Application Server components:

- WAS Application Server - An application server, which can either be a stand-alone server or a server deployed within a Cell.
- WAS Deployment Manager
- WAS Node Agent

Supported software

For information on the software versions that the agent for WebSphere Application Server supports, see the Symantec Operations Readiness Tools (SORT) site: https://sort.symantec.com/agents.

Agent functions

Online

The online function performs the following tasks:

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.
- 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	`Deployment Manager WAS_HOME\profile\ServerProfile\binDir\startManager.bat`
Node Agent	`WAS_HOME\profile\ServerProfile\binDir\startNode.bat`
Application Server	`WAS_HOME\profile\ServerProfile\binDir\startServer.bat`

Offline

The offline function performs the following tasks:

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.

- 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\profile\ServerProfile\binDir\stopManager.bat`
Node Agent	`WAS_HOME\profile\ServerProfile\binDir\stopNode.bat`
Application Server	`WAS_HOME\profile\ServerProfile\binDir\stopServer.bat`

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:

- PidFile based monitoring checks for the for the existence of process running with Pid equal to Pid present in PidFile.
  - If the agent finds such process running then it further searches the command line for that process. The search string includes the values specified in resource attributes ServerName, WAS_NODE, WAS_CELL, ServerProfile, and WAS_HOME. If the search is successful, agent proceeds to SecondLevel monitoring if enabled.
  - If the PidFile is not present or if the agent does not finds a process running with Pid specified in PidFile, agent proceeds to process based monitoring.

  Location of the PidFile is determined using resource attributes as follows: *WAS_HOME*\profile*ServerProfile*\log*ServerName**ServerName*.pid
- Process based 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_CELL, WAS_NODE, and ServerName.
- If second-level monitoring is enabled (if SecondLevelMonitor > 0), the monitor agent function performs a deeper, more thorough state check of the WebSphere Application Server. Second-level monitoring uses the IBM-supplied utility program serverStatus.bat. 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 agent function executes serverStatus.bat during each monitor interval. If SecondLevelMonitor is set to 3, the monitor agent function executes serverStatus.bat every third monitor interval. This mechanism lets you control the system load generated by monitoring.

The serverStatus.bat 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 agent function executes a custom monitor program specified in the MonitorProgram attribute. 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 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 15.

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_CELL, 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 agent for WebSphere Application Server
- Installing the VCS agent for WebSphere Application Server
- Removing the VCS agent for WebSphere Application Server

Before you install the agent for WebSphere Application Server

Ensure that you install and configure Symantec Cluster Server before installing the agent for WebSphere Application Server.

Installing the VCS agent for WebSphere Application Server

Use the Product Installer to install the agent for WebSphere Application Server.

Note: Ensure that you have uninstalled the previous version of this agent, if installed.

To install the VCS agent for WebSphere Application Server

1. Log on to any node in the cluster.

   Ensure that the logged on user has the domain administrative privileges.

2. Download the Agent Pack from the Symantec Operations Readiness Tools (SORT) site: https://sort.symantec.com/agents.

   You can download the complete Agent Pack tar file or the individual agent tar file.

3. Uncompress the file to a temporary location.

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

Windows 2003	`cd1\windows\w2k3\application\websphere_agent\`*`vcs_version`*`\`*`version_agent`*`\websphere_agt.5.1.0.0-GA_w2k3`

Windows 2008 (x64)	For VCS 5.1:
	`cd1\windows\w2k8x64\vcs\application\websphere_agent\5.1\`*`version`*`_agent\websphere_agt.`*`version`*`-GA_w2k8X64\Pkgs`
	For VCS 6.0 and later:
	`cd1\windows\w2k8x64\vcs\application\websphere_agent\`*`vcs_version`*`\`*`version`*`_agent\Pkgs`

Windows 2012 (x64)	`cd1\windows\w2k12x64\vcs\application\websphere_agent\6.1\`*`version`*`_agent\Pkgs`

5. Double-click **vrtsvcswebsphere.msi**.

   Follow the instructions that the install program provides, to complete the installation of the Symantec High Availability agent for WebSphere.

Removing the VCS agent for WebSphere Application Server

Perform the following procedure to uninstall the agent for WebSphere Application Server from a cluster. Perform these steps while the cluster is active.

To uninstall the VCS agent for WebSphere Application Server

1. Ensure that all clustered VCS resources are offline.
2. From the cluster, remove all the resources that use the agent for WebSphere Application Server.
3. Perform the following steps on each node from which you want to uninstall the agent. Ensure that you have a user with administrative privileges.
   - Click **Start > Settings > Control Panel**.
     - On Windows 2008: Navigate to **Programs and Features**
     - On Windows 2008R2/2012: Navigate to **Programs**>**Programs and Features**
   - From the list of programs, select **vrtsvcswebsphere.msi**.
4. Click **Change/Remove**.
5. Follow the instructions that the uninstall program provides, to complete the uninstallation of the agents for WebSphere Application Server.

Chapter 3

Configuring the agent for WebSphere Application Server

This chapter includes the following topics:

- About configuring the agent for WebSphere Application Server
- WebSphere Application Server agent attributes
- Uniquely identifying WebSphere Application Server instances
- Executing a custom monitor program
- Service group configuration options
- Important considerations while configuring the agent

About configuring the agent for WebSphere Application Server

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

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
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.bat. 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.bat 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 20. Type and dimension: string-scalar Default: "" Example: server1

Table 3-1 Required attributes *(continued)*

Required attribute	Description
ServerProfile	Contains the server profile name of the WebSphere Server instance or complete path to the WebSphere Application Server profile. If Profile is installed at the non-default location, provide the complete path to the WebSphere Application Server Profile. In case if profile is installed at the default location, profile name is sufficient. This attribute is applicable to the WebSphere version 6.0 and later, and it 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: C:\Users\administrator\Downloads\AppSrv04_Custom_Location
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 standalone 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 Windows 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 Administrator account, but you may use any account. If User is not set to Administrator, the user name must be synchronized across the systems within the cluster. Type and dimension: string-scalar Default: "" Example: Administrator

Table 3-1 Required attributes *(continued)*

Required attribute	Description
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. The *binDir*\setupCmdLine.bat file also resides at this location. 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 20. **Note:** Both WAS_HOME and WAS_ND_HOME are defined as WAS_HOME in the standard environment file setupCmdLine.bat, which is supplied with WebSphere. Type and dimension: string-scalar Default: "" Example: C:\Program Files\IBM\WebSphereNET\AppServer
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 20. Type and dimension: string-scalar Default: "" Example: was51c1n2
Domain	Specifies the Windows domain name to which the specified user belongs. If the attribute value for User does not belong to a Windows domain, use VCS localization settings to specify the local computer name for each system. Type and dimension: string-scalar Default: No default value Example: USER-DOMAIN
Password	Password for the user. Use the vcsencrypt -agent command to encrypt the password. If you are using the VCS GUI, the GUI automatically encrypts the password. Refer to the VCS documentation for more information about VCSEncrypt. Type and dimension: string-scalar Default: No default value

Table 3-1 Required attributes *(continued)*

Required attribute	Description
WAS_CELL	WebSphere Cell Name of the WebSphere Server instance. This is the network name assigned to the WebSphere Cell. A Cell is a logical group of WebSphere Nodes belonging to the same administrative domain. In a single administrative domain every server has the same CELL name. If there are more than one administrative domains or installations of WebSphere on the system, then the CELL name for these domains is different. All WebSphere Servers, regardless of ServerType, are assigned a Cell name. Type and dimension: string-scalar Default: "" Example: HOST01CELL01

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 monitor program. See "Executing a custom monitor program" on page 20. Type and dimension: string-scalar Default: "" Example: C:\server1\bin\mymonitor.bat
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"
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_CELL, 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_CELL , 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.

Executing a custom monitor program

The monitor function executes a custom monitor program to perform a user-defined WebSphere Application Server state check.

The monitor function executes the MonitorProgram if the following conditions are true:

- The specified utility is a valid executable file.
- The first level process check indicates that the WebSphere Application Server is online.
- The SecondLevelMonitor attribute is either set to 0 or 1, and the second level check indicates that the WebSphere Application Server is online.
- The SecondLevelMonitor attribute is set to greater than 1, but the second level check is deferred for this monitoring cycle.

The monitor operation interprets the program exit code as follows:

110 or 0	WebSphere Application Server is ONLINE
100 or 1	WebSphere Application Server is OFFLINE
99	WebSphere Application Server is UNKNOWN
Any other value	WebSphere Application Server is UNKNOWN

To ensure that the custom monitor program is always available to the agent application, recommends storing the file in a shared directory that is available on an online node.

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

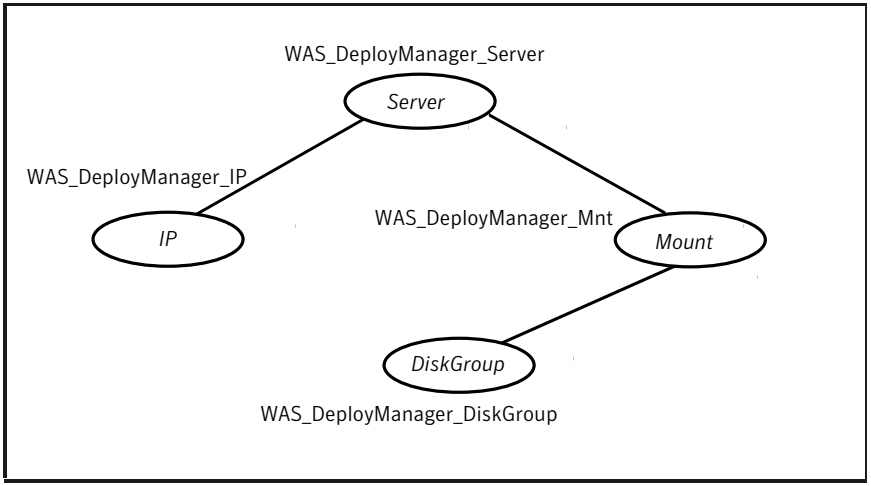


Figure 3-2 depicts a service group that manages a Node Agent Server.

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

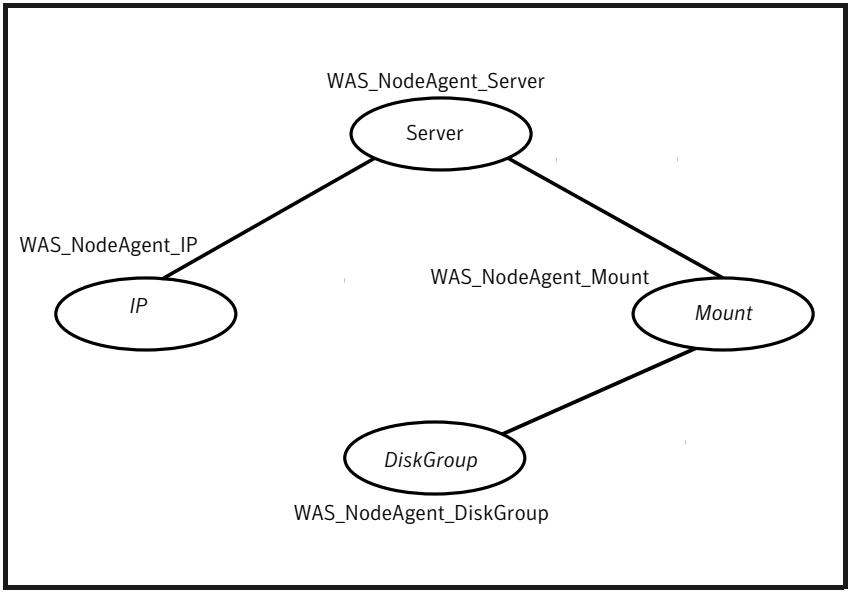
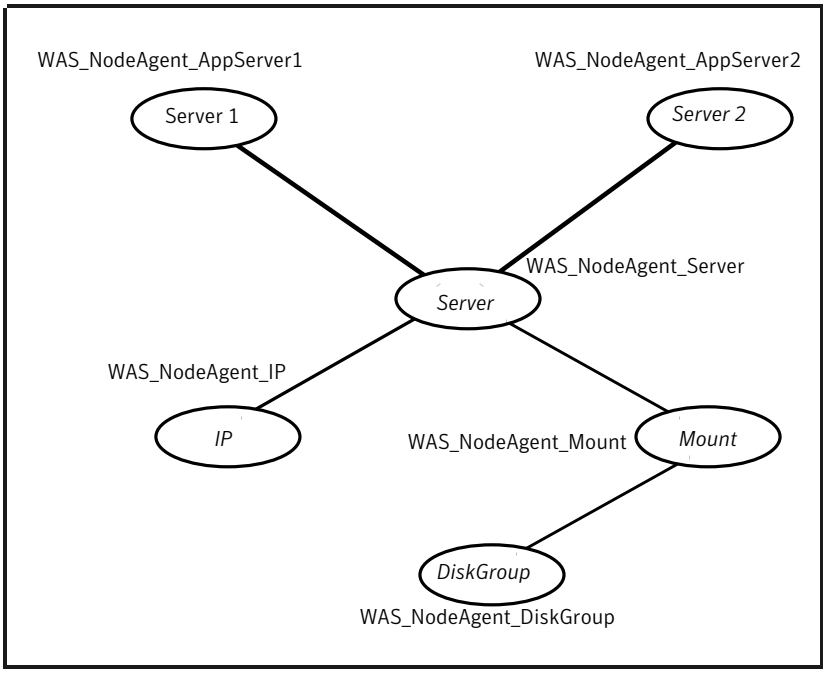


Figure 3-3 depicts a service group that controls a Node Agent Server and its two managed Application Servers.

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



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.

Chapter 4

Configuring the service groups for WebSphere Application Server

This chapter includes the following topics:

- Configuring service groups for WebSphere Application Server

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 (for example, 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 are 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 does not run using the root account, obtain a dedicated Windows account for the WebSphere Application Server. Refer to the description of User attribute for important instructions and requirements to create the account.

Creating service group and supporting resources

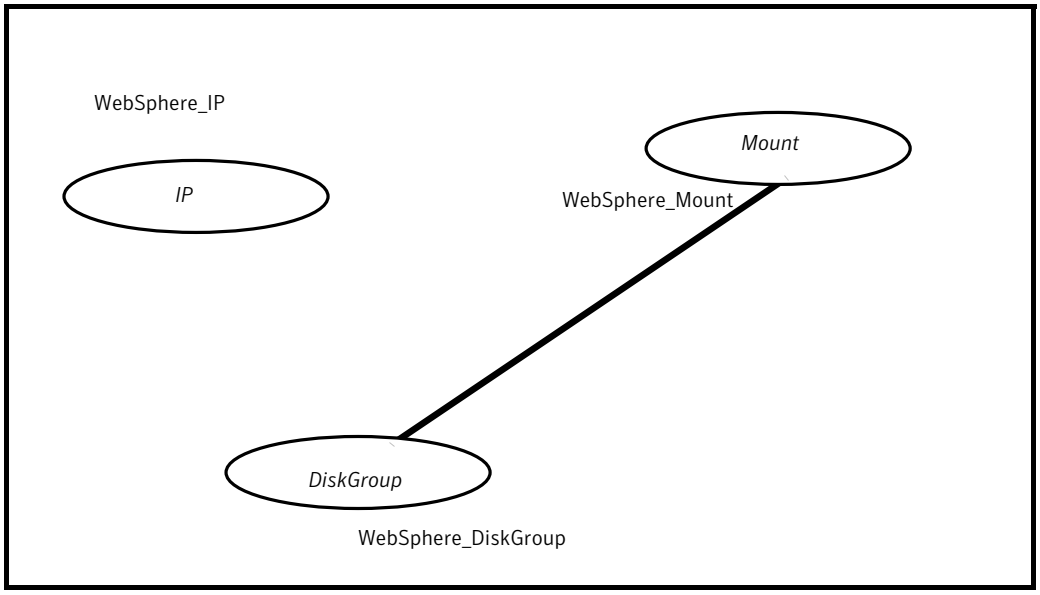
First create a service group on a cluster to contain the resources supporting the WebSphere node.

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
C:\Program Files\IBM\WebSphere	Top level directory under which all WebSphere nodes are installed.
C:\Program Files\IBM\WebSphere\cell1	Subdirectory under which all WebSphere nodes assigned to cell1 are installed.

C:\Program Files\IBM\WebSphere\cell1\depmgr	Subdirectory is the mount point for the shared disk resource dedicated to the Deployment Manager instance supporting cell1.
C:\Program Files\IBM\WebSphere \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.
C:\Program Files\IBM\WebSphere \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.
C:\Program Files\IBM\WebSphere \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.

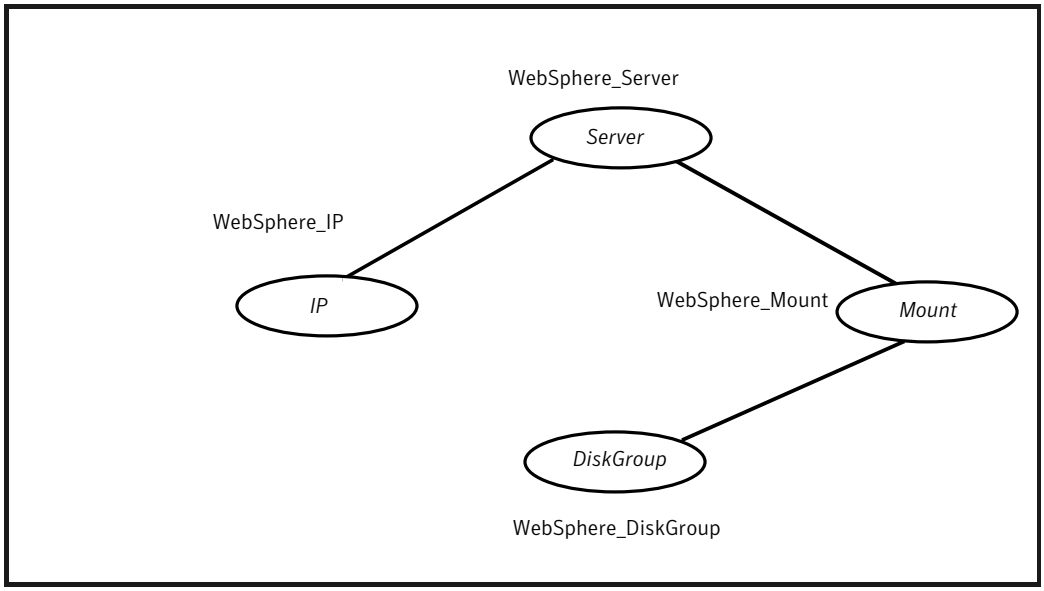
Placing the WebSphere Application Server under cluster control

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
- Starting the WebSphere Application Server outside a cluster
- Reviewing error log files

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, refer to SORT site: https://sort.symantec.com/agents

Starting the WebSphere Application Server 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 independent of the cluster framework. Refer to the cluster documentation for information about disabling a resource. You can then restart the WebSphere Application Server outside the cluster framework.

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

A sample procedure to start a WebSphere Application Server 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_CELL	Host04Cell01
WAS_NODE	was60c1dmsol
WAS_HOME	C:\Program Files\IBM\WebSphere\v60\cell1\depmgr
ServerProfile	Dmgr01

3. Go to directory C:\Program Files\IBM\WebSphere\v60\cell1\depmgr\profiles\Dmgr0\bin.
4. Using the startManager.bat script, start the Deployment Manager.

   C:\Program Files\IBM\WebSphere\v60\cell1\depmgr\profiles\Dmgr01\bin\startManager.bat
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_CELL	Host04Cell01
WAS_HOME	C:\Program Files\IBM\WebSphere\v60\cell1\node\node1
ServerProfile	Default

3. Go to specified directory C:\Program Files\IBM\WebSphere\v60\cell1\node1\profiles\AppSrv\bin.
4. Using the startNode.bat script, start the Node Agent:

   C:\Program Files\IBM\WebSphere\v60\cell1\node1\profiles\AppSrv\bin\startNode.bat
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 the WebSphere Application Server or the agent for WebSphere Application Server, use the error log files described in this section to investigate the problems. Contact Symantec support for more information.

Reviewing VCS log files

In case of problems while using the agent for WebSphere Application Server, you can also access the VCS engine log file for more information about a particular resource.

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

Reviewing agent log files

In case of problems while using the agent for WebSphere Application Server , you can access the agent log files for WebSphere Application Server for more information. The agent saves the output of all the agent operation processes in the c:\program files\veritas\cluster server\log folder. The format of the log file is WebSphere_X.log.

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 is produced. Symantec recommends that you must localize the ResLogLevel attribute for 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 VCS engine output log file. Use the time noted in Step 4 and Step 6 to diagnose the problem.

   Contact Symantec support for more help.

Appendix A

Sample Configurations

This appendix includes the following topics:

- About the sample configuration for the agent for WebSphere Application Server
- Sample WebSphere Application Server agent type definition
- Sample configuration
- Sample service group configuration for the agent for WebSphere Application Server

About the sample configuration for the agent for WebSphere Application Server

The sample configuration 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, refer to the *Symantec Cluster Server Bundled Agents Reference Guide*.

Sample WebSphere Application Server agent type definition

A sample agent type definition is shown as follows:

```
type WebSphere (
 static i18nstr ArgList[] = { ResLogLevel, State, IState,
                              ServerName, User, Domain, Password,
                              WAS_NODE, WAS_HOME, WAS_CELL,
                              ServerProfile, ServerType, StartOptions,
```

```
                                StopOptions, MonitorProgram,
                                SecondLevelMonitor }
 str ResName
 str ResLogLevel = INFO
 str ServerName
 str User
 str Domain
 str Password
 str WAS_NODE
 str WAS_HOME
 str WAS_CELL
 str ServerProfile
 str ServerType
 str StartOptions
 str StopOptions
 str MonitorProgram
 int SecondLevelMonitor
)
```

Sample configuration

A sample main.cf file is shown as follows:

```
include "types.cf"
include "WASTypes.cf"
include "WebSphereTypes.cf"

cluster accwin5 (
 UserNames = { admin = bQRjQLqNRmRRpZRlQO }
 Administrators = { admin }
 )

system sys_A (
 )

system sys_B (
 )

group Grp_WebSphere (
 SystemList = { sys_A = 0 , sys_B = 1}
 )

 IP Res_IP (
```

```
  Critical = 0
  Device = hme0
  Address = "10.136.228.11"
  NetMask = "255.255.248.0"
 )

MountV Res_mount (
   Critical = 0
   MountPath = "C:\\Program Files\\IBM\\WebSphereNET\\APPServer"
   VolumeName = was_log_vol
   VMDGResName = Res_DiskGroup
   ForceUnmount = ALL
   )

VMDg Res_DiskGroup (
   Enabled = 0
   DiskGroupName = Res_DiskGroup
   DGGuid = c844de2f-efbc-431f-b6dd-9a5abc6ece55
 )

WebSphere Res_NodeAgent (
   Critical = 0
   ServerName = nodeagent
   User = administrator
   Domain = ISV-DOMAIN
   Password = ftlRitI
   WAS_NODE = Host04node03
   WAS_HOME = "C:\\Program Files\\IBM\\WebSphereNET\\AppServer"
   WAS_CELL = Host04Cell01
   ServerProfile = AppSrv01
   ServerType = NodeAgent
   StartOptions = "-replacelog -trace"
 )

WebSphere res_DeploymentManager (
   Critical = 0
   ResLogLevel = TRACE
   ServerName = dmgr
   User = Administrator
   Domain = ISV-DOMAIN
   Password = ftlRitI
   WAS_NODE = accnet04CellManager01
   WAS_HOME = "C:\\Program Files\\IBM\\WebSphereNET\\AppServer"
```

```
   WAS_CELL = Host04Cell01
   ServerProfile = Dmgr01
   ServerType = DeploymentManager
   StartOptions = "-trace -replacelog"
 )

WebSphere Res_ApplicationServer1 (
   Critical = 0
   ServerName = server1
   User = administrator
   Domain = ISV-DOMAIN
   Password = ftlRitI
   WAS_NODE = Host04node03
   WAS_HOME = "C:\\Program Files\\IBM\\WebSphereNET\\AppServer"
   WAS_CELL = Host04Cell01
   ServerProfile = AppSrv01
   ServerType = ApplicationServer
   StartOptions = "-replacelog -trace"
 )

WebSphere Res_ApplicationServer2 (
   Critical = 0
   ServerName = server2
   User = administrator
   Domain = ISV-DOMAIN
   Password = ftlRitI
   WAS_NODE = Host04node03
   WAS_HOME = "C:\\Program Files\\IBM\\WebSphereNET\\AppServer"
   WAS_CELL = Host04Cell01
   ServerProfile = AppSrv01
   ServerType = ApplicationServer
   StartOptions = "-trace -replacelog"
   MonitorProgram = "C:\\monitor.bat"
   SecondLevelMonitor = 3
 )

Res_NodeAgent requires res_DeploymentManager
res_DeploymentManager requires Res_IP
res_DeploymentManager requires Res_mount
Res_ApplicationServer1 requires Res_NodeAgent
Res_ApplicationServer2 requires Res_NodeAgent
Res_mount requires Res_DiskGroup
```

```
// resource dependency tree
//
// group Grp_WebSphere
// {
// WebSphere Res_ApplicationServer1
//     {
//     WebSphere Res_NodeAgent
//         {
//         WebSphere res_DeploymentManager
//             {
//             IP Res_IP
//             MountV Res_mount
//                 {
//                 VMDg Res_DiskGroup
//                 }
//             }
//         }
//     }
// WebSphere Res_ApplicationServer2
//     {
//     WebSphere Res_NodeAgent
//         {
//         WebSphere res_DeploymentManager
//             {
//             IP Res_IP
//             MountV Res_mount
//                 {
//                 VMDg Res_DiskGroup
//                 }
//             }
//         }
//     }
// }
```

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 21.

```
group WASCell1DM (
        SystemList = { sysa = 0, sysb = 1, sysc = 2 }
        )
        DiskGroup WASCell1DM_dg (
                DiskGroup = was51c1dm
                )

        IP WASCell1DM_ip (
                Device = hme0
                Address = "10.136.228.11"
                NetMask = "255.255.248.0"
                )

        Mount WASCell1DM_mnt (
                 Critical = 0
                 MountPath = "C:\\Program Files\\IBM\\
                                  WebSphereNET\\APPServer"
                 VolumeName = was_log_vol
                 VMDGResName = WASCell1DM_dg
                 ForceUnmount = ALL
                )

        WebSphere Res_ApplicationServer1 (
                 Critical = 0
                 ServerName = server1
                 User = administrator
                 Domain = ISV-DOMAIN
                 Password = ftlRitI
                 WAS_NODE = Host04node03
                 WAS_HOME = "C:\\Program Files\\IBM\\
                                  WebSphereNET\\APPServer"
                 WAS_CELL = Host04Cell01
                 ServerProfile = AppSrv01
                 ServerType = ApplicationServer
                 StartOptions = "-replacelog -trace"
                )

        Res_ApplicationServer1 requires WASCell1DM_ip
```

```
        Res_ApplicationServer1 requires WASCell1DM_mnt
        WASCell1DM_mnt requires WASCell1DM_dg


group WASCell1Node1 (
        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 (
                Critical = 0
                MountPath = "C:\\Program Files\\IBM\\
                                WebSphereNET\\APPServer"
                VolumeName = was_log_vol
                VMDGResName = Res_DiskGroup
                ForceUnmount = ALL
                )

        WebSphere Res_NodeAgent (
                Critical = 0
                ServerName = nodeagent
                User = administrator
                Domain = ISV-DOMAIN
                Password = ftlRitI
                WAS_NODE = Host04node03
                WAS_HOME = "C:\\Program Files\\IBM\\
                                WebSphereNET\\APPServer"
                WAS_CELL = Host04Cell01
                ServerProfile = AppSrv01
                ServerType = NodeAgent
                StartOptions = "-replacelog -trace"
                )

        WebSphere Res_ApplicationServer1 (
               Critical = 0
```

```
                ServerName = server1
                User = administrator
                Domain = ISV-DOMAIN
                Password = ftlRitI
                WAS_NODE = Host04node03
                WAS_HOME = "C:\\Program Files\\IBM\\
                                WebSphereNET\\APPServer"
                WAS_CELL = Host04Cell01
                ServerProfile = AppSrv01
                ServerType = ApplicationServer
                StartOptions = "-replacelog -trace"
                )

        WebSphere Res_ApplicationServer2 (
                Critical = 0
                ServerName = server2
                User = administrator
                Domain = ISV-DOMAIN
                Password = ftlRitI
                WAS_NODE = Host04node03
                WAS_HOME = "C:\\Program Files\\IBM\\
                                WebSphereNET\\APPServer"
                WAS_CELL = Host04Cell01
                ServerProfile = AppSrv01
                ServerType = ApplicationServer
                StartOptions = "-trace -replacelog"
                MonitorProgram = "C:\\monitor.bat"
                SecondLevelMonitor = 3
                 )

        Res_ApplicationServer2 requires Res_NodeAgent
        WASCell1Node1_mnt requires WASCell1Node1_dg
        Res_NodeAgent requires WASCell1Node1_ip
        Res_NodeAgent_was requires WASCell1Node1_mnt
        Res_ApplicationServer1 requires Res_NodeAgent
                )

group WASAppSrvr1 (
        SystemList = { sysa = 0, sysb = 1, sysc = 2 }
        )

        DiskGroup WASAppSrvr1_dg (
                DiskGroup = was51c1n1
```

```
        )


Mount WASAppSrvr1_mnt (
        Critical = 0
        MountPath = "C:\\Program Files\\IBM\\
                        WebSphereNET\\APPServer"
        VolumeName = was_log_vol
        VMDGResName = Res_DiskGroup
        ForceUnmount = ALL
        )

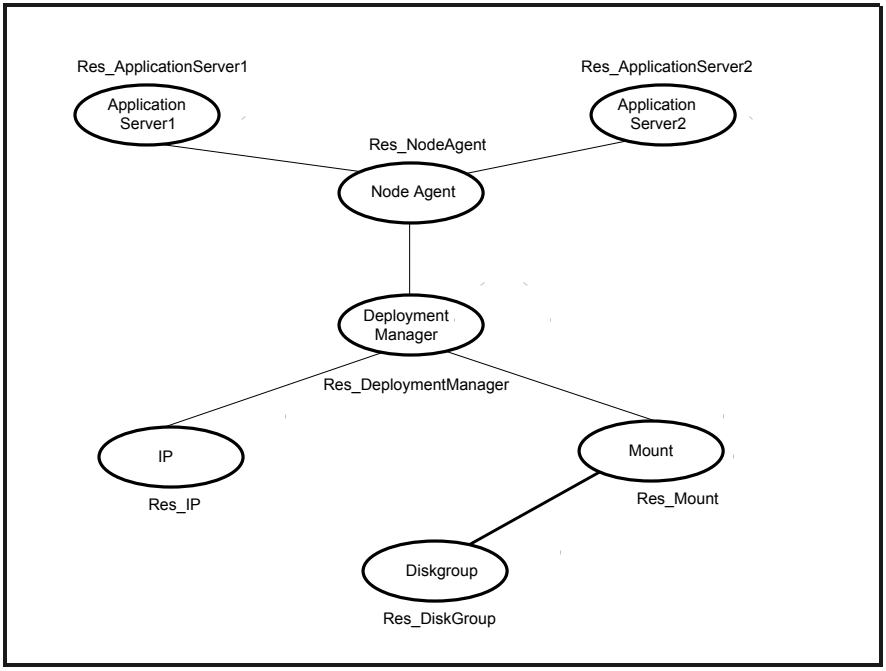

WebSphere Res_ApplicationServer1 (
       Critical = 0
       ServerName = server1
       User = administrator
       Domain = ISV-DOMAIN
       Password = ftlRitI
       WAS_NODE = Host04node03
       WAS_HOME = "C:\\Program Files\\IBM\\
                       WebSphereNET\\APPServer"
       WAS_CELL = Host04Cell01
       ServerProfile = AppSrv01
       ServerType = ApplicationServer
       StartOptions = "-replacelog -trace"
       )


WASAppSrvr1_mnt requires WASAppSrvr1_dg
Res_ApplicationServer1 requires WAS51AppSrvr1_ip
Res_ApplicationServer1 requires WAS51AppSrvr1_mnt
       )
```

Sample service group configuration for the agent for WebSphere Application Server

Figure A-1 depicts a typical service group configuration.

Figure A-1 Service group configuration



Index

A

agent
- configuring service groups 25
- installing in VCS environment 12
- removing in VCS environment;uninstalling in VCS environment 13
- sample configuration 35
- sample service group configuration 42
- type definition 34

agent attributes
- MonitorProgram 19
- ResLogLevel 16
- SecondLevelMonitor 16
- ServerName 16
- ServerProfile 17
- ServerType 17
- StartOptions 19
- StopOptions 19
- User 17
- WAS_HOME 18
- WAS_NODE 18

agent function 9
- clean 11
- monitor 10
- offline 9
- online 9

U

uniquely identifying WebSphere Application server instances 20