

Veritas High Availability Agent 5.0 for WebSphere Application Server Installation and Configuration Guide

Windows 2000, Windows 2003

Veritas High Availability Agent 5.0 for WebSphere Application Server Installation and Configuration Guide

Copyright © 2006 Symantec Corporation. All rights reserved.

5.0

Symantec, the Symantec logo, Veritas 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.

THIS 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” and “commercial computer software documentation” as defined in FAR Sections 12.212 and DFARS Section 227.7202.

Symantec Corporation
20330 Stevens Creek Blvd.
Cupertino, CA 95014
www.symantec.com

Printed in the United States of America.

Third-party legal notices

Third-party software may be recommended, distributed, embedded, or bundled with this Symantec product. Such third-party software is licensed separately by its copyright holder. All third-party copyrights associated with this product are listed in the accompanying release notes.

Windows is a registered trademark of Microsoft Corporation.

Technical support

For technical assistance, visit <http://support.veritas.com> and select phone or email support. Use the Knowledge Base search feature to access resources such as TechNotes, product alerts, software downloads, hardware compatibility lists, and our customer email notification service.

Contents

Chapter 1	Introducing the Veritas Agent for WebSphere Application Server	
	Supported software	3
	About the agent for WebSphere Application Server	4
	Functions that the WASMon.exe utility performs	5
Chapter 2	Configuring VCS Resources for High Availability	
	Configuring a resource for a WAS component	7
Chapter 3	Troubleshooting the Veritas Agent for WebSphere Application Server	
	Using correct software and operating system versions	11
	Troubleshooting WAS components in a high availability environment	12
	Reviewing log files	12
	Using VCS log files	12
	Using WebSphere Application Server components' log files	12
Appendix A	Sample configurations	
	Sample service group configuration	16
	Sample resource configuration	18
Index		23

Introducing the Veritas Agent for WebSphere Application Server

Welcome to the Veritas high availability agent for WebSphere Application Server.

This guide describes the procedures to configure VCS resources of type Process, to provide high availability for WebSphere Application Server components. The guide assumes the reader understands the primary components and basic functionality of Veritas Cluster Server. It also assumes a basic understanding of the WebSphere Application Server architecture and its configuration options.

This chapter contains the following sections:

- [Supported software](#)
- [About the agent for WebSphere Application Server](#)

Supported software

The Veritas Cluster Server Agent for WebSphere Application Server is supported in the following environments:

Environment	Supported Versions
Veritas Cluster Server	4.2, 4.3
Operating Systems	Windows 2000 Server Windows 2003 Server (32-bit)
WebSphere Application Server	5.1, 6.0

About the agent for WebSphere Application Server

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

To provide high availability for the WAS components in the VCS environment, you must first configure VCS resources of type Process. These configured resources can then provide high availability for the WebSphere Application Server components in the VCS environment.

The files that you must use to configure the VCS resources and provide high availability for the WAS components are:

- `WASMon.exe`—This utility can clean and monitor WebSphere Application Server components.
- `timeout_set.vbs`—While configuring a VCS Process resource, this script is used to set the values of the `CleanScriptTimeout` and `OfflineScriptTimeout` attributes for the Process resource. These attributes are set in the registry.
- `timeout_read.vbs`—This script is used to get the `OfflineScriptTimeout` and `CleanScriptTimeout` values for a VCS Process resource, if these values are set in the registry using the `timeout_set.vbs` script.
- `timeout_unset.vbs`—This script is used to remove registry entries for a configured VCS Process resource.

These files are present on the agent pack CD:

Windows Directory version

2000	<code><cd_mount>/windows/w2k/application/websphere_agent/4.3/5.0_agent/websphere5_agt.5.0-GA_w2k</code>
2003	<code><cd_mount>/windows/w2k3/application/websphere_agent/4.3/5.0_agent/websphere5_agt.5.0-GA_w2k3</code>

Refer to the following section for details about configuring VCS resources:

- [“Configuring a resource for a WAS component”](#) on page 7.

Functions that the WASMon . exe utility performs

The WASMon . exe utility performs the following functions:

- **monitor**
- **kill**

The command to execute these functions is as follows:

```
WASMon monitor|kill <WAS_HOME> <CellName> <NodeName>  
<ServerName> [-trace]
```

monitor

The `monitor` function monitors the availability of the WAS server process. The `monitor` function returns **110** if there is a process running in the system whose command line conforms to the following pattern.

- The command line begins with `<WAS_HOME>`.
- The process command line contains the string 'java'.
- The process command line contains `<CellName>` bounded by spaces.
- The process command line contains `<NodeName>` bounded by spaces.
- The process command line contains a space immediately followed by `<ServerName>` at the end of the command line, or the process command line contains `<ServerName>` bounded by spaces.

Example Command Line:

```
C:\was60\srvt1/java/bin/java  
-Xbootclasspath/p:C:\was60\srvt1/java/jre/lib/ext/ibmorb.  
jar;C:\was60\srvt1/java/jre/lib/ext/ibmext.jar  
-Dwas.status.socket=1827 -classpath  
C:\was60\srvt1/profiles/default/properties;C:\was60\srvt1/  
profiles/default/properties;C:\was60\srvt1/lib/bootstrap.jar;C:\was60\srvt1/  
lib/j2ee.jar;C:\was60\srvt1/lib/lmproxy.jar;C:\was60\srvt1/  
lib/urlprotocols.jar -Xms50m -Xmx256m  
-Dws.ext.dirs=C:\was60\srvt1/java/lib;C:\was60\srvt1/profi  
les/default/classes;C:\was60\srvt1/classes;C:\was60\srvt1/  
lib;C:\was60\srvt1/installedChannels;C:\was60\srvt1/lib/ex  
t;C:\was60\srvt1/web/help;C:\was60\srvt1/deploytool/itp/pl  
ugins/com.ibm.etools.ejbdeploy/runtime  
-Dcom.ibm.itp.location=C:\was60\srvt1/bin  
-Djava.util.logging.configureByServer=true  
-Dibm.websphere.preload.classes=true  
-Duser.install.root=C:\was60\srvt1/profiles/default  
-Dwas.install.root=C:\was60\srvt1  
-Djava.util.logging.manager=com.ibm.ws.bootstrap.WsLogMa  
nager -Ddb2j.system.home=C:\was60\srvt1/cloudscape
```

```
-Dserver.root=C:\was60\srvt1/profiles/default-Djava.security.auth.login.config=C:\was60\srvt1/profiles/default/properties/wsjaas.conf  
-Djava.security.policy=C:\was60\srvt1/profiles/default/properties/server.policy com.ibm.ws.bootstrap.WSLauncher  
com.ibm.ws.runtime.WsServer  
C:\was60\srvt1/profiles/default\config accnet04Node01Cell  
was60node_server1
```

If no process is found satisfying the above conditions the monitor function returns 100.

Example Usage:

```
c:\webspheremon\wasmon monitor c:\was60\srvt1 was60node  
accnet04Node01Cell server1
```

kill

The kill function forcibly kills the WAS server process.

The kill function identifies the process running in the system whose command line satisfies the pattern described in the monitor section and terminates the process. If termination is successful, the kill function returns 0. If termination is unsuccessful, the kill function returns 1.

If there is no process running in the system whose command line satisfies the pattern, the kill function again returns 0.

Example Usage:

```
c:\webspheremon\wasmon kill c:\was60\srvt1 was60node  
accnet04Node01Cell server1
```

Configuring VCS Resources for High Availability

To provide high availability for WebSphere Application Server components in the VCS environment, you must first configure the VCS resources of type Process.

This chapter covers the steps that you must perform to configure VCS resources. It contains the following sections:

- [Configuring a resource for a WAS component](#)

Configuring a resource for a WAS component

For a WAS component, you must configure a VCS Process resource using the following procedure.

To configure a WebSphere Application Server resource

- 1 Create and configure a VCS Service Group that consists of a Lanman, an IP address, a mount point directory, and disk group resources.
Refer to the cluster documentation for details about a Service Group.
- 2 Bring the Service Group online.
- 3 Install the WAS component. Ensure that you select the mount directory that you created in [step 1](#) as the WAS home directory.
 - While installing the WAS component, ensure that the component is not installed as a Windows service.

- While installing the WAS component, ensure that you configure the WAS component to listen on the virtual address of the Lanman resource.
- While configuring the node agent, ensure that it does not restart the application servers that it manages.

Note: The WASMon.exe utility attempts to identify the unique process for the WAS component using the values of `<WAS_HOME>` `<cellname>` `<nodename>` `<servername>`. Hence, ensure that the combination of these values is unique for each WAS component which may run on the same system.

- 4 On each node mentioned in the SystemList attribute of the Service Group, create a local directory, for example, `c:\WebsphereMon`. This directory path must be the same for all the nodes.
- 5 On each node, copy the `WASMon.exe.exe`, `timeout_set.vbs`, `timeout_read.vbs`, and `timeout_unset.vbs` files from the agent pack CD, in to the respective local directory.
- 6 Create a VCS Process resource.
- 7 On all the nodes, run this command:

```
cscript timeout_set.vbs <resource_name> <OfflineScriptTimeout>  
<CleanScriptTimeout>
```

where:

- `<resource_name>` is the name of the VCS Process resource created in [step 6](#).
- `<OfflineScriptTimeout>` is the timeout period set for the stop program in seconds. If you do not specify this parameter, the value is automatically set to 180.
- `<CleanScriptTimeout>` is the timeout period set for the clean program in seconds. If you do not specify this parameter, the value is automatically set to 180.

For example, to set the values of these attributes to 120:

```
cscript timeout_set.vbs was60sg_dmgr 120 120
```

You can test your environment and arrive at suitable values for these timeout attributes.

8 Set the values of these attributes for the VCS Process resource:

Attribute	Value/Example
StartProgram	<p>Use the following WebSphere Application Server script:</p> <pre><WAS_HOME>\profiles\<profilename>\bin\startServer.bat <servername></pre> <p>For the node agent, it is:</p> <pre><WAS_HOME>\profiles\<profilename>\bin\startNode.bat</pre> <p>For example:</p> <pre>c:\WAS60\node1\profiles\profile1\bin\startServer.bat server1</pre>
MonitorProgram	<pre>c:\WebSphereMon\WASMon.exe monitor <WAS_HOME> <cellname> <nodename> <servername></pre> <p>For example:</p> <pre>c:\WebSphereMon\WASMon.exe monitor c:\was60\node1 cell1 node1 server1</pre>
StopProgram	<p>Use a WebSphere Application Server script.</p> <pre><WAS_HOME>\profiles\<profilename>\bin\stopServer.bat <servername></pre> <p>For the node agent, it is:</p> <pre><WAS_HOME>\profiles\<profilename>\bin\stopNode.bat</pre> <p>For example:</p> <pre>c:\WAS60\node1\profiles\profile1\bin\stopServer.bat server1</pre>
CleanProgram	<pre>c:\WebSphereMon\WASMon.exe kill <WAS_HOME> <cellname> <nodename> <servername></pre> <p>For example:</p> <pre>c:\WebSphereMon\WASMon.exe kill c:\was60\node1 cell1 node1 server1</pre>

The VCS Process resource is now configured and ready. To ensure that the resource is working, perform the following steps:

- 1** Attempt the following:
 - Online the VCS Process resource.
 - Offline the VCS Process resource.

- Force clean the Process resource as follows:
 - Specify an incorrect value for StopProgram attribute.
 - Offline the VCS resource.
 - Ensure that the clean function kills the relevant WebSphere Application Server process.

2 Test the switchover operation to ensure that the operation runs successfully.

For a sample service group configuration, refer to “[Sample service group configuration](#)” on page 16 and “[Sample resource configuration](#)” on page 18.

Troubleshooting the Veritas Agent for WebSphere Application Server

This chapter covers issues related to the VCS resources that are configured to provide high availability for WebSphere Application Server components. To resolve issues effectively, refer to the following sections. You may come across unique issues, for which you can contact Symantec support.

This chapter contains the following sections:

- [Using correct software and operating system versions](#)
- [Troubleshooting WAS components in a high availability environment](#)
- [Reviewing log files](#)

Using correct software and operating system versions

To ensure that no issues arise due to incorrect software and operating system versions, refer to “[Supported software](#)” on page 3 for the correct versions of operating system and software to be installed on the resource systems.

Troubleshooting WAS components in a high availability environment

This section covers some of the problems that you may face while using WAS components in a cluster environment.

VCS Process agent kills StopProgram and CleanProgram before completion

If you have not set the values of the OfflineScriptTimeout and CleanScriptTimeout attributes for the VCS Process resource in the Windows registry, this problem occurs. If you do not set these values, the Process agent uses 10 seconds as the default value, which may be insufficient for the stop and clean operations to complete execution.

To resolve the issue, use the `timeout_set.vbs` script to set the values of the OfflineScriptTimeout and CleanScriptTimeout attributes for the Process resource. For more information, refer to [step 7](#) of “[To configure a WebSphere Application Server resource](#)” on page 7.

Reviewing log files

If the configured VCS resource is not working properly, you can review the log files to diagnose the problem.

Using VCS log files

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

The VCS engine log file is in the following location:

```
c:\program files\veritas\cluster server\log\engine_A.txt.
```

For a VCS Process resource, you can view the following Process agent log file:

```
c:\program files\veritas\cluster server\log\Process_A.txt.
```

Using WebSphere Application Server components' log files

If a WebSphere Application Server is facing problems, you can view the server log files to further diagnose the problem.

For WAS v6.0, the log files are located at:

```
<WAS_HOME>\profiles\<profile name>\logs\<server name>
```

Example:

```
c:\was60\node1\profiles\profile1\logs\server1
```


For WAS v5.1, the log files are located at:

`<WAS_HOME>\logs\<server name>`

Example:

`c:\ibm\was51\DeploymentManager\logs\dmgr`

Sample configurations

This appendix describes sample service group configurations and resource configurations for high availability for WebSphere Application Server components.

It contains the following sections:

- [Sample service group configuration](#)
- [Sample resource configuration](#)

Sample service group configuration

Figure A-1 depicts a typical service group configuration.

In this service group configuration:

- Was60sg_node1server1_process is a Process resource used to manage a WAS application server deployed within a cell.
- Was60sg_node1nodeagent_process is a Process resource used to manage a WAS Node agent.
- Was60sg_node1server1_process is a Process resource used to manage a WAS stand-alone Application server.

Figure A-1 Typical service group configuration

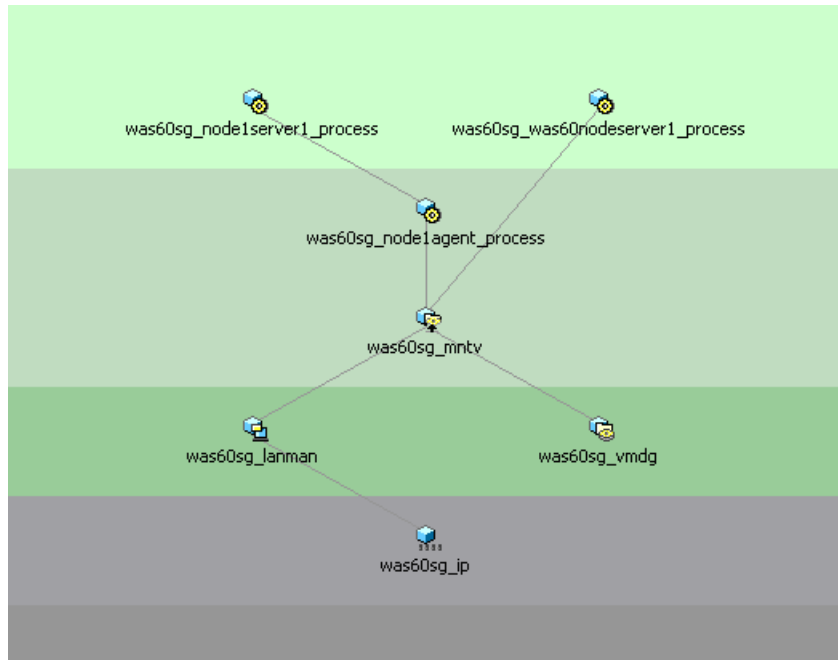
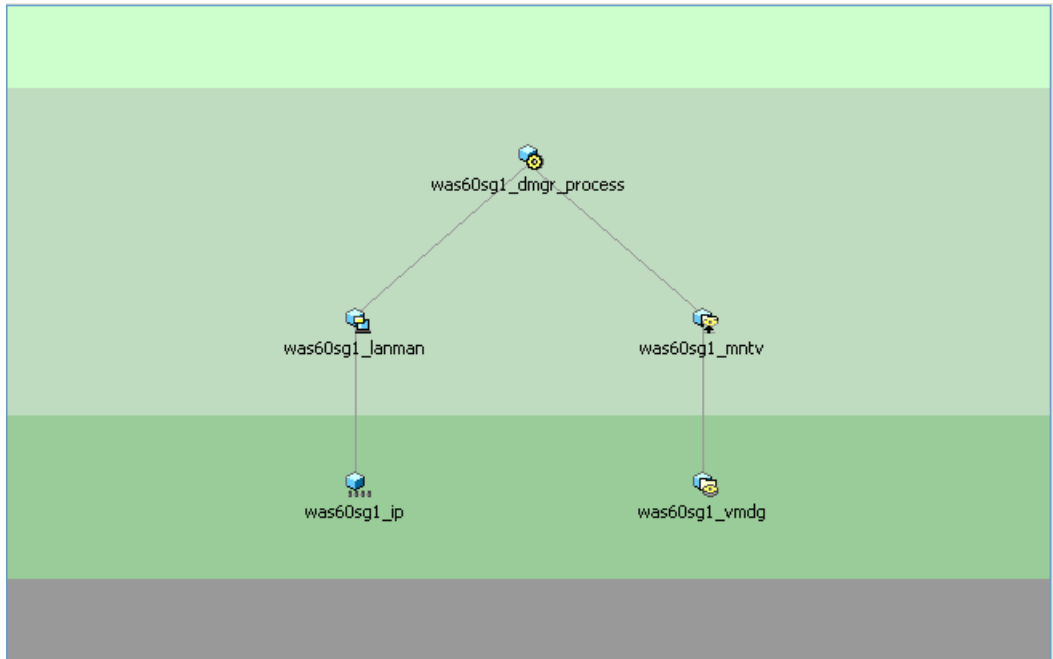


Figure A-2 depicts a service group configuration for a WAS deployment manager.

In the service group shown in this figure, `was60sg1_dmgr_process` is a Process resource used to manage a WAS deployment manager.

Figure A-2 Service group configuration for WAS deployment manager



Sample resource configuration

Figure A-3 shows the configuration for a WAS application server (deployed within a cell) Process resource.

Figure A-3 Sample configuration for a WAS application server

Attribute	Value
• StartProgram	: c:\was60\node1\profiles\profile1\bin\startServer.bat server1
• StopProgram	: c:\was60\node1\profiles\profile1\bin\stopServer.bat server1
• MonitorProgram	: c:\webspheremon\wasmon monitor c:\was60\node1 node1 cell1 server1
• UserName	:
• Password	:
• Domain	:
• MonitorProgramTimeout	: 30
• InteractWithDesktop	: false
• CleanProgram	: c:\webspheremon\wasmon kill c:\was60\node1 node1 cell1 server1
• StartupDirectory	:

Figure A-4 shows the configuration for the WAS node agent Process resource.

Figure A-4 Sample configuration for the WAS node agent Process resource

Attribute	Value
• StartProgram	: c:\was60\node1\profiles\profile1\bin\startNode.bat
• StopProgram	: c:\was60\node1\profiles\profile1\bin\stopNode.bat
• MonitorProgram	: c:\websphere\mon\wasmon monitor c:\was60\node1 node1 cell1 nodeagent
• UserName	:
• Password	:
• Domain	:
• MonitorProgramTimeout	: 30
• InteractWithDesktop	: false
• CleanProgram	: c:\websphere\mon\wasmon kill c:\was60\node1 node1 cell1 nodeagent
• StartupDirectory	:

Figure A-5 shows the configuration for the WAS Standalone Application Server Process resource

Figure A-5 Sample configuration for the WAS Standalone application server Process resource

Attribute	Value
• StartProgram	: c:\was60\srvt1\profiles\default\bin\startServer.bat server1
• StopProgram	: c:\was60\srvt1\profiles\default\bin\stopServer.bat server1
• MonitorProgram	: c:\webspheremon\wasmon monitor c:\was60\srvt1 was60node accnet04Node01Cell server1
• UserName	:
• Password	:
• Domain	:
• MonitorProgramTimeout	: 30
• InteractWithDesktop	: false
• CleanProgram	: c:\webspheremon\wasmon kill c:\was60\srvt1 was60node accnet04Node01Cell server1
• StartupDirectory	:

Figure A-6 shows the configuration for the WebSphere 6.0 Deployment Manager Process resource.

Figure A-6 Sample configuration for the WebSphere 6.0 deployment manager Process resource

Attribute	Value
• StartProgram	: c:\was60_dm\dmnode\profiles\Dmgr01\bin\startServer.bat dmgr
• StopProgram	: c:\was60_dm\dmnode\profiles\Dmgr01\bin\stopServer.bat dmgr
• MonitorProgram	: c:\webspheremon\wasmon monitor c:\was60_dm\dmnode cell1 dmnode dmgr
• UserName	:
• Password	:
• Domain	:
• MonitorProgramTimeout	: 30
• InteractWithDesktop	: false
• CleanProgram	: c:\webspheremon\wasmon kill c:\was60_dm\dmnode cell1 dmnode dmgr
• StartupDirectory	:

Index

A

- about
 - agent for WebSphere Application Server 4
- agent
 - configuring 7
 - functions 5
 - introducing 3
 - sample service group configuration 16
 - troubleshooting 11
- agent configuration
 - configuring a WebSphere Application Server 7
- agent functions
 - kill 6
 - monitor 5

S

- sample configurations 15
- sample service group configuration 16
- supported software 3

T

- timeout_read.vbs 4
- timeout_set.vbs 4
- timeout_unset.vbs 4
- troubleshooting 11

W

- WASMon.exe 4

