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

AIX, HP-UX, Linux, Solaris

5.1

✸ symantec™

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

The software described in this book is furnished under a license agreement and may be used only in accordance with the terms of the agreement.

Agent version: 5.1.0.0

Document version: 5.1.0

Legal Notice

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

Introducing the Veritas High Availability Agent for Oracle Application Server

This chapter includes the following topics:

About the Veritas agent for Oracle 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 clustering features of the Oracle Application Server create a highly scalable, available, and easy-to-manage environment along with a VCS cluster. By clustering Oracle Application Server instances with the Veritas agent for Oracle Application Server, you can ensure application availability. You can also implement disaster recovery using the Global Cluster option.

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

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

The Veritas agent for Oracle Application Server brings the Oracle instances online, monitors the instances, and brings the instances offline. The agent monitors the system processes and server states, and can shutdown the server in case of a failover.

The Veritas agent for Oracle Application Server manages and monitors the following Oracle Application Server programs and instances:

- Oracle Application Server
  The Oracle Application Server is either stand-alone or deployed within an Oracle Container for J2EE (OC4J) cluster. The Agent for Oracle Application Server manages the OPMN, HTTP server (OHS), Web Cache, OC4J components, DCM daemon and so on, on the Oracle Application Server Service Groups.
  See "Oracle Application Server Containers for J2EE (OC4J) Clustering" on page 14.

- Oracle Application Server Infrastructure
  The agent for Oracle Application Server manages the OID, SSO/DAS, and HTTP Server, and the DCM daemon on the Infrastructure Service Group.

- Oracle SOA Suite
  The agent for Oracle Application Server manages the SOA Suite feature added in Oracle AS 10gR3 release by creating the resources for OPMN and OC4J home for Oracle AS10gR3 ORACLE_HOME.

- Oracle Process Monitoring and Notification (OPMN) daemon
  OPMN starts and monitors multiple Oracle Application Server components. The agent for Oracle Application Server operations use OPMN to monitor and manage these components. The agent also monitors and protects the OPMN daemon itself.

- Enterprise Manager control processes
  The Enterprise Manager control processes manage the Enterprise Manager Webconsole. The agent for Oracle Application Server manages the Enterprise Manager Control Console process, such as, iasconsole, dbconsole and agent.

What's new in this agent

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

- Added support for VCS One 2.0

- Added support for Oracle Application Server 10gR3

- Added support for Solaris 10 zones in VCS 5.0 environment

- Integrated with an enhanced version of ACC library

Supported software

The Veritas agent for Oracle Application Server supports the following software versions in a VCS environment:

Veritas Cluster Server	4.0, 4.1, 5.0
ACC Library	5.1.4.0 and later
Operating Systems	■ AIX—5L, 6L on pSeries ■ HP-UX—11i v2 on PA-RISC ■ Linux—Red Hat Enterprise 3.0, 4.0 ■ Solaris—8, 9, 10 on SPARC
Oracle Application Server	Oracle Application Server 10gR1, 10gR2, 10gR3

The Veritas agent for Oracle Application server supports the following software versions in a VCS One environment:

Veritas Cluster Server One	VCS One 2.0 on AIX, HP-UX, Linux, and Solaris
Operating Systems	■ AIX—5L, 6L on pSeries ■ HP-UX—11i v2 on PA-RISC ■ Linux—Red Hat Enterprise 3.0, 4.0 ■ Solaris—8, 9, 10 on SPARC
Oracle Application Server	Oracle Application Server 10gR1, 10gR2, 10gR3

Integrating with the high availability framework of Oracle Application Server

The 10g and later releases of Oracle Application Server include the following additional capabilities for enhancing application availability. The agent for Oracle Application Server in a clustered environment, integrate with these capabilities to further enhance availability.

Oracle Process Monitoring and Notification (OPMN) daemon

OPMN starts and monitors multiple Oracle Application Server components. The agent for Oracle Application Server operations use OPMN to monitor and manage these components. The agent also monitors and protects the OPMN daemon itself.

> **Note:** OPMN is required for starting and stopping the other OAS components. To avoid conflict between the Agent for Oracle Application Server and OPMN, you must configure OPMN such that it does not restart the failed components. Within the clustered environment, OPMN is used only for starting and stopping the Oracle Application Server components.

Oracle Application Server Containers for J2EE (OC4J) Clustering

The Oracle Application Server provides a mechanism to cluster the application server processes so that different processes can share the same workload. You can balance the load of client requests using the Oracle HTTP server, Web Cache, or a third-party load balancer. The Oracle Application Server also provides state replication and configuration management among clustered instances.

The J2EE and Veritas clustering technologies are complimentary and work at different levels:

- OC4J clustering focuses on application objects and session and messaging status. This level of clustering helps you build scalable applications, simplifies management of distributed applications, and reduces the client impact of application server failures.

- Veritas clustering manages and protects the entire application infrastructure that includes the OC4J cluster, file servers, enterprise applications, back end database, and so on. The agent for Oracle Application Server also protects single points of failure in the Oracle Application Server infrastructure, such as the OPMN daemon.

The agent for Oracle Application Server can restart the failed OC4J components on alternate cluster nodes that restores the application. By including Oracle Application Server components in service groups, you can implement site-wide failover with the Global Cluster Option, for rapid disaster recovery purposes.

Oracle Application Server agent functions

The Veritas Agent for OAS performs the following functions:

Online

The online function performs the following tasks:

- Validates that the agent attributes are set properly.

- Verifies that the Oracle Application Server instance is not already online.

- Starts the Enterprise Manager Console Processes by executing the `emctl start` command.

- Starts the OPMN Managed Processes by executing the `opmnctl startproc` command.

The online function exits either after all the processes start successfully, or after the timeout period specified in the OnlineTimeout attribute expires.

Offline

The offline function performs the following tasks:

- Verifies that the Oracle Application Server instance is not already offline.

- Stops the Enterprise Manager Console Processes by executing the `emctl stop` command.

- Stops the OPMN Managed Processes by executing the `opmnctl stopproc` command.

The offline function exits either after all the processes stop successfully, or after the timeout period specified in the OfflineTimeout attribute expires.

Monitor

The monitor function monitors the states of the Oracle Application Server instances running on all nodes within the cluster. The monitor function performs the following tasks:

- The first level check searches for all system processes that must be running for an Oracle Application Server instance. The check executes the process utility (ps) command and searches for the strings that uniquely identify the Oracle Application Server instance. These search strings include the values specified in the ComponentProcessType and User attributes. If the first level check does not find these processes running on the node, the check exits immediately, and reports the instance as OFFLINE.

- If the SecondLevelMonitor attribute is set to greater than 0, the monitor function uses the OPMN and emctl daemon to check the status of the instance. The operation runs the opmnctl and emctl status commands to check the status of the instance. If the instance is online, the second level check returns the status as ONLINE.
  See "Executing second level monitoring" on page 36.

- Depending upon the MonitorProgram attribute, the monitor operation can perform a customized check using a user supplied monitoring utility.
  For details about executing a custom monitor program:

See "Executing a customized monitoring program" on page 37.

Clean

In case of a failure or after an unsuccessful attempt to online or offline an Oracle Application Server instance, the clean operation removes any Oracle Application Server processes remaining in the system. The function performs the following tasks:

- Attempts to shut down the Oracle Application Server instance using the `opmnctl` and `emctl` commands.

- Kills the remaining processes pertaining to this OAS, if the instance does not shut down normally.

Installing, upgrading, and removing the agent for Oracle Application Server

This chapter includes the following topics:

Before you install the Veritas agent for Oracle Application Server

For VCS, do the following:

- Install and configure Veritas Cluster Server.

For more information on installing and configuring Veritas Cluster Server refer to, `Veritas Cluster Server Installation Guide`

- Remove any previous version of this agent.
  To remove the agent,
  See "Removing the agent in a VCS environment" on page 22.

- Install the latest version of ACC Library.
  To install or update the ACC Library package, locate the library and related documentation on the agentpack disc.
  See "Installing the ACC library" on page 19.

For VCS One, do the following:

- Install and configure Veritas Cluster Server One.
  For more information on installing and configuring Veritas Cluster Server refer to, `Veritas Cluster Server One Installation Guide`

- Remove any previous version of this agent.
  To remove the agent,
  See "Removing the agent in a VCS One environment" on page 23.

Prerequisites for installing the agent to support Solaris zones

Ensure that you meet the following prerequisites to install the agent for Oracle Application Server:

- Install the Oracle Application Server to support Solaris zones. For details, refer to note 330016.1 on the metalink.oracle.com site.

- Install and configure the VCS 5.0 environment to support Solaris zones. Refer to the VCS user documentation for details.

- Ensure that you set up the zones and the required system parameters correctly. For details, refer to the Sun documentation.

- Remove any previous version of this agent.

About ACC Library

The operations for the Veritas agent for Oracle 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 Oracle Application Server. The ACC library contains common, reusable functions that perform tasks, such as process identification, logging, and system calls.

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 Oracle 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/oracleas_agent/

vcs_*version*/*version*_agent/pkgs

HP-UX   *cd_mount*/hpux/generic/application/oracleas_agent/

vcs_*version*/*version*_agent/pkgs

Linux   *cd_mount*/linux/generic/
application/oracleas_agent/vcs_*version*/
*version*_agent/rpms

Solaris   *cd_mount*/solaris/*dist_arch*/application/

oracleas_agent/vcs_*version*/*version*_agent/pkgs

Where *dist* is the Solaris distribution and *arch* is the Solaris processor architecture.

3 Install the package.

AIX   `# installp -ac -d VRTSvcsoas9.rte.bff VRTSvcsoas9.rte`

HP-UX   `# swinstall -s 'pwd' VRTSvcsoas9`

Linux   `# rpm -ihv \`
          `VRTSvcsoas9-`*AgentVersion*`-GA_GENERIC.noarch.rpm`

Solaris   `# pkgadd -d . VRTSvcsoas9`

About installing the agent in a VCS One environment

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

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

Following are the commonly used options that installagpack program supports.

-addtypes	Use this option to add the type definition for the agents that are shipped with agent pack installer.
	See "Adding the agent type definitions in a VCS One environment" on page 30.
-rmtypes	Use this option to remove the type definition for the agents that are shipped with agent pack installer.
-responsefile	Use this option to perform automated VCS One High Availability Agents Installation using the system and the configuration information that is stored in a specified file instead of prompting for information.
	The responsefile *response_file* must be a full path name. If -responsefile option is not specified, the response file is automatically generated as installagpack*RANDSTRING*.response, where *RANDSTRING* is a six character string of random alpha-numerals. You must edit the response file to use it for subsequent installations. Variable field definitions are defined within the file.
-rsh	Use this option to specify that rsh and rcp are to be used for communication between systems instead of ssh and scp. This option requires that systems be preconfigured such that rsh commands between systems execute without prompting for passwords or confirmations.

Installing the agent in a VCS One environment

Installing the agent packs involves the following phases:

■ Installing the agent packages

■ Adding the agent resource type definitions
See "Adding the agent type definitions in a VCS One environment" on page 30.

Installing the agent packages

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

To install the Veritas high availability agents

1   Mount the Agent Pack software disc on the client system where you plan to run the installation.

2   Depending on the platform type, navigate to the directory containing the installer for the VCS One agents:

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

3   Start the installagpack program.

    # ./installagpack

4   Enter the name of a client system or client systems where you want to install the agents.

5   Select the agent to install.

6   After the agent is installed, run the installagpack program using the -addtypes option, to add the agent type to the Policy Master configuration.

    # ./installagpack -addtypes

7   Enter the virtual IP address of the Policy Master.

8   Select the agent type you want to add.

9   Review the output as the installation program installs the agent packages.

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

Removing the agent in a VCS environment

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

```
# hares -list Type=OracleAS9
```

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

```
# hatype -delete OracleAS9
```

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 Oracle Application Server from each node in the cluster.

Execute the following command to uninstall the agent:

AIX	`# installp -u VRTSvcsoas9.rte`
HP-UX	`# swremove VRTSvcsoas9`
Linux	`# rpm -e VRTSvcsoas9`
Solaris	`# pkgrm VRTSvcsoas9`

Removing the agent in a VCS One environment

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

To remove the Veritas high availability agents from client systems

1 Mount the Agent Pack software disc on the client system where you plan to run the uninstallagpack program.

2 Depending on the platform type, navigate to the directory containing the uninstaller for the VCS One agents:

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

3 Start the uninstallagpack program.

```
# ./uninstallagpack
```

4 Enter the names of the client systems from which you want to uninstall the agent. The names must be separated by spaces.

5 Run the installagpack program using the -rmtypes option, to remove the agent type from the Policy Master configuration.

```
./installagpack -rmtypes
```

6 Enter the virtual IP address of the Policy Master.

7 Review the output as the program verifies the agent pack that you installed and removes the agent packages.

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

Removing the ACC library

Perform the following steps to remove the ACC library.

To remove the ACC library

1 Ensure that all agents that use ACC library are removed.

2 Run the following command to remove the ACC library package.

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

Upgrading the agent in a VCS environment

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

1 Persistently freeze the service groups that host the application.

```
# hagrp -freeze group -persistent
```

2 Stop the cluster services forcibly.

```
# hastop -all -force
```

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

```
# ps -ef |grep OracleAS9
```

4 Uninstall the agent package from all the nodes.

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

5 Install the new agent on all the nodes.

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

6 Copy the new OracleAS9Types.cf file from the agent's sample conf directory,

| VCS 4.x | /etc/VRTSagents/ha/bin/OracleAS9 |
| VCS 5.0 | /etc/VRTSvcs/conf/sample_OracleAS9 |

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

Note: If you are using Solaris Zones, copy the OracleAS9Types_zones.cf file from the agent's sample conf directory.

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

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

---

8   Start the cluster services.

```
# hastart
```

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

```
# haagent -start OracleAS9 -sys System
```

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

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

Upgrading the agent in a VCS One environment

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

1   Freeze service groups that hosts the application.

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

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

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

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

```
# ps -ef |grep OracleAS9
```

4   Uninstall the agent package from all the nodes.

See "Removing the agent in a VCS One environment" on page 23.

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

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

6   Add the agent types, using the installagpack program.

See "Adding the agent type definitions in a VCS One environment" on page 30.

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

8   Start the clients.

   # `hastart -client`

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

   # `haagent -start OracleAS9 -sys `*`System`*

10 Unfreeze the service groups.

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

Preparing to configure the agent for Oracle Application Server

This chapter includes the following topics:

Importing the agent types files in a VCS environment

To use the agent for Oracle 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_OracleAS9/OracleAS9Types.cf
VCS 5.0	/etc/VRTSagents/ha/conf/OracleAS9/OracleAS9Types.cf
For Solaris zone support	/etc/VRTSagents/ha/conf/OracleAS9/OracleAS9Types_zones.cf

4   Click **Import**.

5   Save the VCS configuration.

The Oracle Application Server configuration is now imported to the VCS engine.

You can now create Oracle 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 in a VCS One environment

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

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

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

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

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

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

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

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

   ■ Mount the Agent Pack software disc.

   ■ Depending on the platform type, navigate to the directory containing the installer for the agents:

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

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

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

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

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

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

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

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

Oracle Application Server agent attributes

Table 3-1 lists the required attributes.

Table 3-1   Required attributes

Attribute	Description
ComponentProcessType	The process type of the Oracle Application Server component, that the resource is managing. The OPMN daemon uses this attribute to start, monitor, and stop the Oracle Application Server instance.
	The attribute values for Oracle Infrastructure components are:
	■ HTTP_Server, for HTTP_Server component type. ■ dcm-daemon, for dcm-daemon component type. ■ OC4J_SECURITY, for OC4J component type. ■ OID, for OID component type.
	The attribute values for Oracle Application Server components are:
	■ HTTP_Server, for HTTP_Server component type. ■ dcm-daemon, for dcm-daemon component type. ■ WebCache and WebCacheAdmin, for WebCache component type. ■ OC4J_Portal, home and OC4J_BI_Forms, for OC4J component type. ■ OC4J_Wireless, for wireless component type. ■ OAD or OSAgent, for Discoverer component type.
	The attribute values for Enterprise Manager components are:
	■ iasconsole, for iasconsole component type. ■ dbconsole, for dbconsole component type. ■ agent, for EMagent component type.
	The attribute values for Oracle Application Server SOA suite components are:
	■ oc4j_soa, for OC4J oc4j_soa component type. This attribute value must match the value that was set during OAS installation. ■ home, for OC4J home component type.
	If you set this attribute to opmn, you can manage the OPMN daemon itself.
	Note: The value of this attribute must match the value that the opmnctl status command returns.
	Type and definition: string-scalar
	Default: ""
	Example: OC4J_Portal

Table 3-1   Required attributes *(continued)*

Attribute	Description
ContainerName	Name of the Solaris zone on which the Oracle Application Server component is running.
	This attribute is applicable for Solaris zones support only.
	Type and definition: string-scalar
	Default: ""
	Example: appsrv_zon1
OracleHome	The absolute path to the Oracle Application Server installation directory. The Agent for Oracle Application Server uses this attribute to locate the programs to be executed.
	By default, Oracle does not provide any standard scripts or environment setup files. All binaries are referenced relative to the OracleHome directory.
	Type and definition: string-scalar
	Default: ""
	Example: /u01/app/oracle/Infrastructure
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 operations.
	Type and dimension: string-scalar
	Default: INFO
	Example: TRACE

Table 3-1 Required attributes *(continued)*

Attribute	Description
User	The UNIX user name used to start, monitor, and stop the Oracle Application Server resource. This user is also the owner of the Oracle Application Server installation. If you specify the MonitorProgram attribute, the security credentials of this user are used to run the specified custom monitor utility.
	You must synchronize the user names across all the systems within the cluster. All user names must resolve to the same UID and the same default shell on each system in the cluster.
	You can define the user locally or in a common repository, such as NIS, NIS+, or LDAP. The agent operations use the getpwname(3C) system call to obtain all UNIX user attributes.
	Type and dimension: string-scalar
	Default: ""
	Example: oracle

Table 3-2 lists the optional attributes

Table 3-2 Optional attributes

Attribute	Description
MonitorProgram	Absolute path name of an external, user-supplied monitor executable. For information about setting this attribute:
	See "Executing a customized monitoring program" on page 37.
	Type and dimension: string-scalar
	Default: ""
	Example: /u01/app/oracle/Infra/opmn/bin/oasmonitor.sh

Table 3-2   Optional attributes *(continued)*

Attribute	Description
SecondLevelMonitor	Used to enable second-level monitoring. Second-level monitoring is a deeper, more thorough state check of the Oracle Application Server instance. The numeric value specifies how often the monitoring routines must run. 0 means never run the second-level monitoring routines, 1 means run routines every monitor interval, 2 means run routines every second monitor interval, and so on. See "Executing second level monitoring" on page 36. **Note:** Exercise caution while setting SecondLevelMonitor to large numbers. For example, if the MonitorInterval is set to 60 seconds and the SecondLevelMonitor is set to 100, then the agent performs SLM every 100 minutes, which may not be as often as intended. For maximum flexibility, no upper limit is defined for SecondLevelMonitor. Type and dimension: integer-scalar Default: 0 Example: 1

Executing second level monitoring

If the SecondLevelMonitor attribute is set to greater than 0, the Agent for Oracle Application Server performs a second level check to monitor the state of the Oracle Application Server instance.

The second-level check uses the Oracle supplied opmnctl and emctl scripts to check the status of the instance.

The OPMN managed components executes the opmnctl status command and parses the status column of the output. The status column displays Alive for online and Down for offline or unknown.

The iasconsole and dbconsole components executes the emctl status <*iasconsole*/*dbconsole*> command and checks the exit code of the command. If the exit code is 0 then the component is online else offline.

For example:

For OPMN

```
bash-3.00$ ./opmnctl status
```

```
Processes in Instance: ashp1.vcshp1.vxindia.veritas.com
---------------------------------------------------------
ias-component process-type pid status
---------------------------------------------------------
OC4J          OC4J_BI_Forms  27684   Alive
OC4J          home           N/A     Down
WebCache      WebCacheAdmin  9234    Alive
WebCache      WebCache       9287    Alive
HTTP_Server   HTTP_Server    27590   Alive
dcm-daemon    dcm-daemon     14416   Alive
LogLeader     logleaderd     N/A     Down
```

Note: If the component is OPMN daemon, the second-level monitor runs the `opmnctl status` command and checks the return code.

For emctl

```
bash-3.00$ ./emctl status dbconsole
Oracle Enterprise Manager 10g Database Control Release 10.1.0.3.0
Copyright (c) 1996, 2005 Oracle Corporation. All rights reserved.
http://ora10g.vxindia.veritas.com:5500/em/console/aboutApplication
Oracle Enterprise Manager 10g Database Control is not running.
bash-3.00$ echo $?
2

bash-3.00$ ./emctl status dbconsole
Oracle Enterprise Manager 10g Database Control Release 10.1.0.3.0
Copyright (c) 1996, 2005 Oracle Corporation. All rights reserved.
http://ora10g.vxindia.veritas.com:5500/em/console/aboutApplication
Oracle Enterprise Manager 10g Database Control is running.
-----------------------------------------------------------------
Logs are generated in directory /ora10g/OraHome_1/ora10g.
vxindia.veritas.com_orcl/sysman/log
bash-3.00$ echo $?
0
```

Executing a customized monitoring program

The monitor function can execute a customized monitoring utility to perform an additional Oracle Application Server state check. The monitor function executes the utility specified in the MonitorProgram attribute, if:

■ The specified utility is a valid executable file.

- The first level process check indicates that the Oracle Application Server instance is online.

- The SecondLevelMonitor attribute is either set to 0 or 1, and the second level check indicates that the Oracle Application Server instance is online.

- The SecondLevelMonitor attribute is set to greater than 1, but the second level check is deferred for this monitoring cycle.

The monitor function interprets the utility exit code as follows:

110 or 0	Oracle Application Server instance is online
100 or 1	Oracle Application Server instance is offline
99	Oracle Application Server instance is unknown
Any other value	Oracle Application Server instance is unknown

To ensure that the customized utility is always available to the agent, Symantec recommends storing the file in a shared directory that is available on the online node.

Configuring the Oracle Application Server resources for Solaris zone support

To enable the Agent for Oracle Application Server to support Solaris 10 zones, ensure that you perform these configuration steps:

- For each Oracle Application Server resource, set the ContainerName attribute to the name of the local zone in which the Agent for Oracle Application Server is installed.

- VCS must be installed on the global zones, on all the nodes in the cluster.

- Ensure that the required disk group and mount point are available.

- The zone in which the Oracle Application Server must be installed and configured correctly. You must be able to boot the zone on any of the physical nodes in the cluster.

- Configure the virtual IP address on the Oracle Application Server as the zone IP address.

- You must install the Agent for Oracle Application Server under the /opt directory. This directory must be available on the local zone as a lofs file system.

For sample Service Group configurations:

Configuring the service groups for Oracle Application Server

This chapter includes the following topics:

■ Configuring an Oracle Application Server in a clustered environment

Configuring an Oracle Application Server in a clustered environment

The Veritas agent for Oracle Application Server can manage and monitor the following Oracle Application Server programs and components:

■ Oracle Process Monitoring and Notification daemon (OPMN) daemon

■ Oracle Application Server Infrastructure

■ Standalone or clustered Oracle Application Servers

■ Oracle SOA Suite

■ Enterprise Manager control processes

You can cluster the Oracle Application Server resources using various methods. Symantec recommends the following procedure to cluster Oracle Application Server resources in a clustered environment.

Perform the following steps in the order presented below:

■ Allocating shared disk resources

■ Creating disk group, volume, and file system

-
-
-
-
-
-
-

Allocating shared disk resources

Symantec recommends installing each Oracle Application Server instance on a separate and dedicated shared disk resource. Obtain a shared disk resource for each instance to be clustered.

Creating disk group, volume, and file system

Create a disk group, volume, and file system on a shared disk resource that is allocated for an Oracle Application Server.

Obtaining dedicated virtual IP address and DNS name

Obtain unique virtual addresses and DNS names for each Oracle Application Server instance. These addresses and names are required to support the Oracle Application Server IP network configuration.

Each Oracle Application Server node uses the virtual IP address and DNS name dedicated to the node, irrespective to the cluster system hosting the node.

Obtaining dedicated user accounts

If the Oracle Application Server instance does not run using a root account, you must assign a dedicated User account for the instance.

Creating service group and supporting resources

Perform the following tasks:

- Create a service group that can contain all required resources to support the Oracle Application Server instance.

- Create appropriate cluster resources and link the shared disk resource and network objects that were previously under the cluster control.

- Bring the service group online to test the configuration.

An example of an Oracle Application Server service group configuration follows.

Installing the Oracle Application Server software

Ensure that the shared disk and network resources are online. You can now install the Oracle Application Server and the Oracle Application Server Infrastructure software. Consider the following steps while installing the software:

See "Avoiding port conflicts" on page 43.

See "Installing the Oracle Application Server software on a shared disk file system" on page 43.

Avoiding port conflicts

If other applications are running on the target system, the Oracle Application Server installer detects the busy ports during installation and uses other free ports for installing the component. If the ports are free, a port conflict may occur after installation. Symantec recommnends that you pre-configure dedicated port numbers for each Oracle Application Server instance before starting the installation.

You can specify the port numbers in the staticports.ini file. A sample staticports.ini file exists on the Oracle Application Server CD under Disk1/Stage/Response directory. Copy this file to an appropriate location and modify the port numbers for the components. Then run the installer using the appropriate command line arguments. Refer to the Oracle documentation for more information.

For example, for installing an Oracle Application Server Infrastructure component, run the installer:

```
./runInstaller
\oracle.iappserver.infrastructure:s_\StaticPorts=<full pathname to
the staticports.ini file >
```

Refer to the Oracle Application Server installation guide for information on using the static ports feature for installation.

Installing the Oracle Application Server software on a shared disk file system

Follow the instructions in the Oracle Application Server product documentation to install the software. Ensure that the installation program installs the software

on the shared disk file system previously established for this Oracle Application Server instance.

■ While installing the Oracle Application Server Infrastructure, select High Availability and Replication in the Oracle Universal Installer: Select Configuration Options dialog box.

■ In the Oracle Universal Installer: Select High Availability or Replication Option, select Virtual host to configure all components to use a virtual hostname.

■ When prompted for the virtual hostname, enter the dedicated virtual hostname allocated to the node on which you are performing the installation.

■ While installing SOA Suite, set the environment variable VIRTUAL_HOST_NAME to the desired virtual hostname, before you proceed with the installation.

Configuring virtual hostname on a standalone or clustered Oracle Application Server

The installer for installing a standalone or clustered Oracle Application Server does not provide the feature to configure the virtual hostname for the node. You must configure the virtual hostname manually.

For manual configuration, you must do the following:

■ Enable the virtual IP address assigned to the Oracle Application Server installation.

■ Associate a hostname with the virtual IP address in the DNS or local host file.

■ Stop all Oracle Application Server processes except OPMN.

■ Change the physical hostname in all the configuration files for the components installed on the server, to the virtual hostname. You must change all such occurrences where the physical hostname is used as a hostname instead of part of an instance name or other key.

As an example, follow these steps to perform the manual configuration. Refer to the Oracle documentation for exact details about these configuration files.

■ To modify the HTTP Server

■ To modify WebCache

■ To modify the portal

■ To modify wireless

■ To modify OracleBI Discoverer

■ To modify dbconsole

■ To verify all changes

To modify the HTTP Server

1   In the following files, change all such occurrences where the physical hostname is used as a hostname instead of part of an instance name or other key:

■ $ORACLE_HOME/Apache/Apache/conf/httpd.conf

■ $ORACLE_HOME/Apache/Apache/conf/mod_oc4j.conf

■ $ORACLE_HOME/Apache/Apache/htdocs/index.html

2   Update the DCM repository:

```
# $ORACLE_HOME/dcm/bin/dcmctl updateConfig –ct ohs
```

3   Re-register mod_osso as a partner application:

■ Verify that the OracleHome variable is set correctly.

■ Set LD_LIBRARY_PATH as equal to $ORACLE_HOME/lib32.

■ Set LD_LIBRARY_PATH_64 as equal to $ORACLE_HOME/lib.

■ Run this command to register mod_osso as a partner application:

```
# $ORACLE_HOME/jdk/bin/java -jar
$ORACLE_HOME/sso/lib/ossoreg.jar -site_name <virtual
hostname> -mod_osso_url http://<virtual hostname>
:<http port no> -config_mod_osso TRUE
-oracle_home_path <$ORACLE_HOME> -u <user>
-config_file
$ORACLE_HOME/Apache/Apache/conf/osso/osso.conf
-admin_infocn=orcladmin
```

To modify WebCache

1   Make this change in the $ORACLE_HOME/webcache/webcache.xml file:

```
<CACHE NAME="cachename" ORACLEHOME=<ORACLE_HOME>
"HOSTNAME="<hostname>" VOTES="1" CAPACITY="30"
WCDEBUGON="NO">
```

2   Copy the CACHE NAME complete tagset and paste it below the existing one.

3 Change the HOSTNAME to the physical hostname of the clustered machine.

For example,

```
<CACHE NAME="<cachename>" ORACLEHOME="<OracleHome>"
HOSTNAME="<HostName1>" VOTES="1" CAPACITY="30" WCDEBUG ON="NO">
........
........
........
 </CACHE>

 <CACHE NAME="<cachename>" ORACLEHOME="<OracleHome>"
HOSTNAME="<HostName2>" VOTES="1" CAPACITY="30" WCDEBUG ON="NO">
........
........
........
 </CACHE>
```

4 Change the hostname to virtual hostname in the
$ORACLE_HOME/webcache/docs/welcome.html file.

To modify the portal

1 Change all occurrences of physical hostname to virtual hostname in the
$ORACLE_HOME/portal/conf/cache.xml file.

Note: If the physical hostname is part of an instance name or any other key,
do not change the hostname.

---

2 Follow these steps to retrieve the password for the portal schema from OID:

■ From the Infrastructure ORACLE_HOME, run this command:
`$ORACLE_HOME/bin/oidadmin`

■ Log in to OID as cn=orcladmin.

■ Open the Oracle Directory Manager screen.

■ In the left panel, navigate to **Entry management > cn=OracleContext >
cn=Products > cn=IAS > cn=IAS Infrastructure Databases >
orclReferenceName=Global_SID**.

■ Select OrclResourceName=PORTAL.
In the right panel, take note of the password that is displayed in the
orclpasswordattribute field.

3   Register the URL changes with the OracleAS portal:

```
# cd $ORACLE_HOME/portal/conf (OracleHome of the Oracle
Application Server)

# ./ptlconfig –dad portal -pw <password> -wc -site
```

4   Restart the components:

$ORACLE_HOME/opmn/bin/opmnctl startall

5   Follow these stepts to update the Oracle Application Server Wireless Portal Service URL reference:

- Log in to the portal using the virtual hostname:
  http://<virtual hostname>:port/pls/portal>

- Enter the administrator user name and password.

- Click **Administrator**.

- Under Services, click **Global Settings**.

- Click the **Mobile** tab.

- Change the Wireless Portal Service URL entry to use the virtual hostname.

- Click **Apply**.

To modify wireless

1   In the ORACLE_HOME/wireless/config/iaswcfg.xml file, change all occurrences of physical hostname to virtual hostname.

2   Register the wireless device portal with Oracle Single Sign-On:

```
# cd $ORACLE_HOME/wireless/bin
# reRegister.sh \ http://<virtual_hostname>:port/ptg/rm \
$ORACLE_HOME "cn=orcladmin"
```

To modify OracleBI Discoverer

◆   Change all occurrences of physical hostname to virtual hostname in the following files:

Note: If the physical hostname is part of an instance name or any other key, do not change the hostname.

- ORACLE_HOME/discoverer/config/configuration.xml

- HOME/j2ee/properties/oc4j_bi_forms.properties

■ ORACLE_HOME/j2ee/OC4J_BI_Forms/config/oc4j.properties

To modify dbconsole

◆ During the installation setup, in the $ORACLE_HOME/bin/emctl file set the following:

■ ```
ORACLE_HOSTNAME = <virtual hostname>
ORACLE_SID=<Oracle SID set at Installation time>
export ORACLE_HOSTNAME ORACLE_SID
```

**To verify all changes**

1  Log in to the following portal using the virtual hostname:

   **http://<virtual hostname>:port/pls/portal**

   ---

   **Note:** All the following steps are not applicable to every installation. Perform the relevant steps only. For details, refer to the Installing in High Availability Environments section in the Oracle Application Server Installation Guide.

   ---

2  Enter administrator user name and password.

3  Click **Administrator**.

4  Under Services, click **Global Settings**.

5  Click **Configuration**.

6  Verify that the host entry in Default JPDK Instance is based on the virtual hostname.

7  Click the **Cache** tab.

8  Verify that the host name entry in Web Cache Host Settings is based on the virtual hostname.

9  Click the **Mobile** tab.

10  Verify that the OracleAS 10g Wireless Portal Service URL entry in the OracleAS 10g Wireless Information section is based on virtual hostname.

11  Click the **Administrator** tab.

12  Under Services, click **Search Settings**.

13  Verify that the hostname in Oracle Text Base URL is based on the virtual hostname.

# Configuring the OPMN daemon to work in a cluster

The agent for Oracle Application Server is designed to manage and monitor individual Oracle Application Server components. Therefore, you must configure the OPMN daemon to not take any automatic action if it detects a failure of a component under cluster control.

For all Oracle Application Server components, disable the restart-on-death option in the opmn.xml Oracle Application Server configuration file. This file is typically located in the $ORACLE_HOME/opmn/conf directory.

The restart-on-death parameter is defined in the process-set section of each Oracle Application Server component. For components like OC4J that have multiple instances, you must make this change for the process-set section under each process-type.

An example to show the changes for the HTTP_Server component follows.

```
<ias-component id="HTTP_Server">
<process-type id="HTTP_Server" module-id="OHS">
<module-data>
<category id="start-parameters">
<data id="start-mode" value="ssl-disabled"/>
</category>
</module-data>
<restart timeout="0" retry="0"/>
<process-set id="HTTP_Server" restart-on-death="false"
numprocs="1"/>
</process-type>
</ias-component>
An example to show the changes for two process types for
an OC4J component follows.
<ias-component id="OC4J">
<dependencies>
<OID infrastructure="true"/>
</dependencies>
<process-type id="home" module-id="OC4J" status="disabled">
<module-data>
<category id="start-parameters">
<data id="java-options" value="-server -
Djava.security.policy=$ORACLE_HOME/j2ee/home/config/java2.policy -
Djava.awt.headless=true"/>
</category>
<category id="stop-parameters">
<data id="java-options" value="-
```

```
Djava.security.policy=$ORACLE_HOME/j2ee/home/config/java2.policy -
Djava.awt.headless=true"/>
</category>
</module-data>
<start timeout="600" retry="2"/>
<stop timeout="120"/>
<restart timeout="720" retry="2"/>
<port id="ajp" range="3301-3400"/>
<port id="rmi" range="3201-3300"/>
<port id="jms" range="3701-3800"/>
<process-set id="default_island" restart-on-death="false"
numprocs="1"/>
</process-type>
<process-type id="OC4J_SECURITY" module-id="OC4J">
<environment>
<variable id="DISPLAY"
value="oasinfra.vxindia.veritas.com:0.0"/>
<variable id="LD_LIBRARY_PATH"
value="/oasinfra/ohome/lib32:/oasinfra/ohome/lib"/>
</environment>
<module-data>
<category id="start-parameters">
<data id="java-options" value="-server -
Djava.security.policy=/oasinfra/ohome/j2ee/OC4J_SECURITY/config/jav
a2.policy -
Djava.awt.headless=true -Xmx512m "/>
<data id="oc4j-options" value="-properties"/>
</category>
<category id="stop-parameters">
<data id="java-options" value="-
Djava.security.policy=/oasinfra/ohome/j2ee/OC4J_SECURITY/config/jav
a2.policy -
Djava.awt.headless=true"/>
</category>
</module-data>
<start timeout="900" retry="2"/>
<stop timeout="120"/>
<restart timeout="720" retry="2"/>
<port id="ajp" range="3301-3400"/>
<port id="rmi" range="3201-3300"/>
<port id="jms" range="3701-3800"/>
<process-set id="default_island" restart-on-death="false"
numprocs="1"/>
```

```
</process-type>
</ias-component>
```

You must make changes in the opmn.xml file for each Oracle Application Server component that is to be placed under cluster control.

Placing the Oracle Application Server component under cluster control

Create necessary cluster resources using the Agent for Oracle Application Server and place each Oracle Application Server instance under cluster control.

To properly define the resource attributes:

See "Oracle Application Server agent attributes" on page 32.

For a sample service group configurations:

See "Sample service group configuration" on page 58.

Troubleshooting the agent for Oracle Application Server

This chapter includes the following topics:

- Using correct software and operating system versions
- Meeting prerequisites
- Configuring Oracle Application Server resources
- Configuration checks for Solaris zone support
- Starting the Oracle Application Server instance 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:

See "Supported software" on page 13.

Meeting prerequisites

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

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

See "Before you install the Veritas agent for Oracle Application Server" on page 17.

Configuring Oracle Application Server resources

Before using an Oracle Application Server resource, ensure that you configure the resource properly. For a list of resource types with which you can configure all Oracle Application Server resources, refer to the agent attributes.

Configuration checks for Solaris zone support

If you have configured the Agent for Oracle Application Server to support Solaris zones, ensure that you have followed all the configuration steps:

■ Prerequisites for enabling Solaris zone support
  See "Before you install the Veritas agent for Oracle Application Server" on page 17.

■ Importing the types.cf file for Solaris zone support
  See "Importing the agent types files in a VCS environment" on page 29.

■ Configuring the Oracle Application Server resources for Solaris zone support
  See "Configuring the Oracle Application Server resources for Solaris zone support" on page 38.

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

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

To restart the Oracle Application Server instance outside the framework

1   Log in as a user with administration rights for the Oracle Application Server instance:

    ```
    # su - user
    ```

2   Execute these commands to set the user environment:

    ```
    $ ORACLE_HOME=<OracleHome>
    $ PATH=$ORACLE_HOME/bin:$ORACLE_HOME/opmn/bin:$PATH
    $ LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH
    $ export ORACLE_HOME PATH LD_LIBRARY_PATH
    ```

3   Run the opmnctl command to start the component:

    ```
    $ opmnctl startproc process-type=<ComponentProcessType>
    ```

    For example, to start an OID component, run this command:

    ```
    $ opmnctl startproc process-type=OID
    ```

Ensure that the Oracle Application Server instance starts successfully outside the cluster framework. If the instance is working properly, attempt to start the instance inside the cluster framework.

Reviewing error log files

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

Using Oracle Application Server log files

If an Oracle Application Server is facing problems, you can access the server log files to further diagnose the problem. The Oracle Application Server log files are located in the $ORACLE_HOME/opmn/logs directory. Each component's start and stop process has a corresponding log file in this directory. For example, $ORACLE_HOME/opmn/logs/HTTP_Server~1.

Reviewing cluster log files

In case of problems while using the agent for Oracle 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.

Using trace level logging

The ResLogLevel attribute controls the level of logging that is written in a cluster log file for each Oracle 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   Test the identified resource. The function reproduces the problem that you are attempting to diagnose.

5   Set the ResLogLevel attribute back to INFO for the identified resource:

    ```
    # hares -modify Resource_Name ResLogLevel INFO -sys SysA
    ```

6   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.

Sample Configurations

This appendix includes the following topics:

- About sample configurations for the agent for Oracle Application Server
- Sample agent type definition
- Sample service group configuration
- Sample Oracle Application Server resource configuration
- Sample service group configurations for Solaris zone support

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

Sample agent type definition

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 OracleAS9Types.cf file in the /etc/VRTSvcs/conf/config cluster configuration directory.

An excerpt from this file follows.

```
type OracleAS9
(
```

```
static str ArgList[] = { ResLogLevel, User, OracleHome,
ComponentProcessType, SecondLevelMonitor, MonitorProgram }
str ResLogLevel = INFO
str User
str OracleHome
str ComponentProcessType
int SecondLevelMonitor
str MonitorProgram
)
```

Sample service group configuration

The following service group configuration demonstrates how an Oracle Application Server environment can be configured to spread the processing load over several systems in a clustered configuration. The example shows all the required services linked within a cluster. This environment clusters all the services, from the front-end to the back-end, in a classic multi-tier architecture that could support a wide range of applications.

Figure A-1 shows a six-node clustered environment is shown below. A business application (CRM, Finance, ERP, and so on) is spread across the six nodes:

■ Web servers on the first two nodes

■ Middle-tier application servers on the next two nodes

■ Database and Oracle Infrastructure components running on the fifth node

■ The sixth node is a hot-standby node to which any of the application services could fail over in the event of a fault.

Figure A-1   Overview of Service Group

Each application service is installed on shared disk file systems provided by a SAN, which is essential to enable each application service to run on any node in the cluster.

This example demonstrates a common HA strategy called an "N+1 configuration". In an N+1 configuration, a single dedicated system is available to take over any services in the event of the failure of any of the other hosts in the cluster. The N+1 configuration has the advantage of allowing the application environment to operate at full CPU bandwidth in the event of any single system failure. For this to be true, all of the systems must be configured with similar processing capacity (speed, number of CPUs and memory).

Figure A-2 shows the startup dependencies for all service groups. You must always start the Oracle Application Server Infrastructure Service Group before any other Oracle Application Server Service Group. Therefore, the Service Group dependency is set to online global (soft). This dependency setting prevents the Oracle Application Server Service Groups from stopping if the Infrastructure Service Group restarts, faults, or is directed to switch over to another system in the cluster.

This Service Group containing the production database for application environments is independent of the other Service Groups.

Figure A-2   Dependency settings for Oracle Application Server Service Group

Figure A-3 shows a service group that includes the Oracle database and the Oracle Listener. This group represents the standard production database service that is part of the classic three-tier application environment.

The Agent for Oracle Application Server is not used in this sample Service Group. Use the agent to manage the Oracle Database and Listener processes.

Figure A-3   Oracle Database and Listener Service Group

Figure A-4 shows a Service Group that manages Oracle Application Server
Infrastructure components. Since the Agent for Oracle Application Server uses
the OPMN daemon to start, monitor, and stop the Oracle Application Server
components, you must start the OPMN component before starting any other
component.

Ensure that you start the Oracle Distributed Configuration Management (DCM)
component and the OC4J_SECURITY component used for Single Sign-on (SSO)
after the Oracle Internet Directory (OID) component. The diagram also illustrates
that the OID component should not be started until the Oracle Infrastructure
Database containing the OID data is available.

Figure A-4   Oracle Application Server Infrastructure Service Group

Figure A-5 and Figure A-6 show Service Groups that manage the Oracle Application Server HTTP server and Webcache services.

Since the OPMN component is configured to start before the other Oracle Application Server components, Having multiple service groups for this function offers greater capacity as these normally run on dedicated host systems.

Figure A-5   Sample 1: Service Group for Oracle Application Server HTTP server and Webcache services

Figure A-6   Sample 2: Service Group for Oracle Application Server HTTP server
                    and Webcache services

Figure A-7 and Figure A-8 show Service Groups that manage the Oracle Application Server OC4J Application Servers. Since the OPMN component is configured to start before the other Oracle Application Server components, Having multiple service groups for this function offers greater capacity as these normally run on dedicated host systems.

The example shows an HTTP server within the Service Group so that the users can access the Oracle Application Server via HTTP. An HTTP module, mod_oc4j redirects network requests to the appropriate Oracle Application Server instance.

Figure A-7   Sample 1: Service Group for Oracle Application Server OC4J
                  Application Server

Figure A-8   Sample 2: Service Group for Oracle Application Server OC4J
                  Application Server

Sample Oracle Application Server resource configuration

A sample excerpt from the main.cf file is shown below.

```
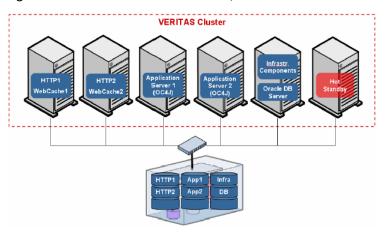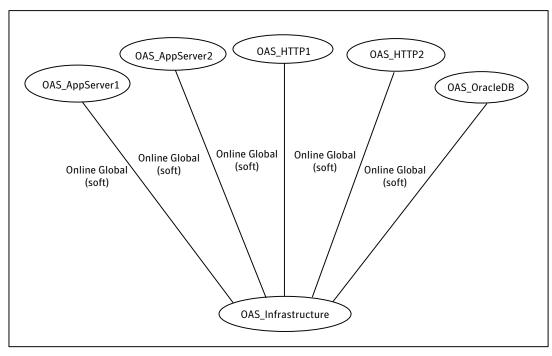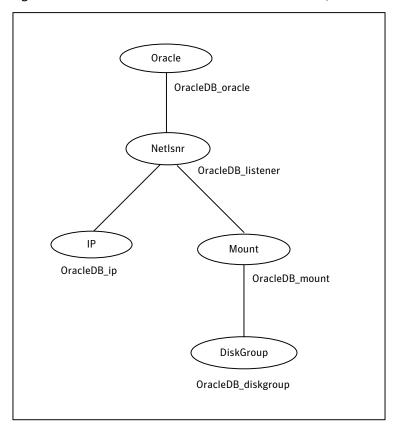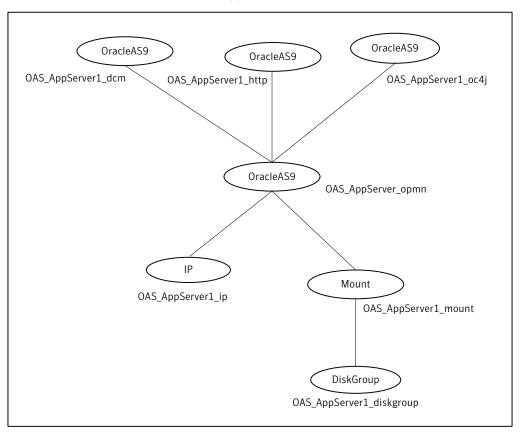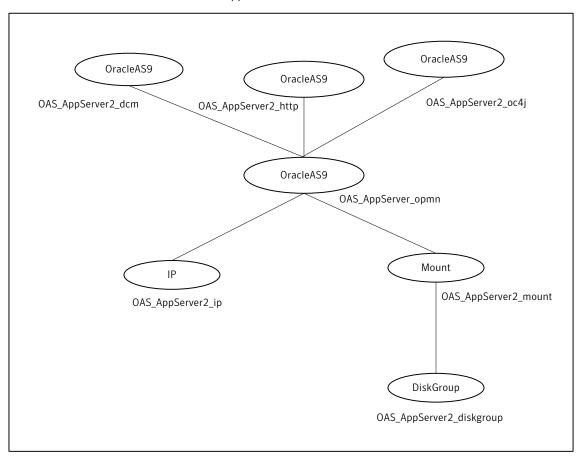group OraAS10gInfrastructure (
 SystemList = { vcssx074 = 0, vcssx075 = 1 }
 )

 DiskGroup OraAS10g_dg (
  DiskGroup = oradg
```

```
)

IP OraAS10g_IP (
 Device = bge0
 Address = "10.209.73.86"
 NetMask = "255.255.252.0"
 )

Mount OraAS10g_mnt (
 MountPoint = "/ora10g"
 BlockDevice = "/dev/vx/dsk/oradg/oravol"
 FSType = vxfs
 MountOpt = rw
 FsckOpt = "-y"
 )

OracleAS9 OraAS10g_HTTPSrvr (
 User = oracle
 OracleHome = "/ora10g/OraHome_1"
 ComponentProcessType = HTTP_Server
 )

OracleAS9 OraAS10g_OC4JSecurity (
 Enabled = 0
 User = oracle
 OracleHome = "/ora10g/OraHome_1"
 ComponentProcessType = OC4J_SECURITY
 )

OracleAS9 OraAS10g_OID (
 Enabled = 0
 User = oracle
 OracleHome = "/ora10g/OraHome_1"
 ComponentProcessType = OID
 )

OracleAS9 OraAS10g_OPMN (
 Critical = 0
 User = oracle
 OracleHome = "/ora10g/OraHome_1"
 ComponentProcessType = OPMN
 SecondLevelMonitor = 5
 )
```

```
OracleAS9 OraAS10g_dbconsole (
 User = oracle
 OracleHome = "/ora10g/OraHome_1"
 ComponentProcessType = dbconsole
 )

OracleAS9 OraAS10g_dcm (
 Enabled = 0
 User = oracle
 OracleHome = "/ora10g/OraHome_1"
 ComponentProcessType = dcm-daemon
 )

OracleAS9 OraAS10g_iasconsole (
 User = oracle
 OracleHome = "/ora10g/OraHome_1"
 ComponentProcessType = iasconsole
 )

OraAS10g_mnt requires OraAS10g_dg
OraAS10g_HTTPSrvr requires OraAS10g_OPMN
OraAS10g_OPMN requires OraAS10g_mnt
OraAS10g_OC4JSecurity requires OraAS10g_OID
OraAS10g_OID requires OraAS10g_OPMN
OraAS10g_OPMN requires OraAS10g_IP
OraAS10g_dbconsole requires OraAS10g_IP
OraAS10g_dcm requires OraAS10g_OID
OraAS10g_iasconsole requires OraAS10g_OPMN
```

Sample service group configurations for Solaris zone support

This section includes sample service groups with Solaris zone support.

Figure A-9 shows a service group for an Oracle Application Server Infrastructure with a metadata repository and resources for OPMN , OID, OC4J_Security, and HTTP.

Figure A-9   Sample Service Group with Solaris zone support

Figure A-10 shows a sample service group with Oracle HTTP Server, OC4J, and DCM running in a zone.

Figure A-10   Sample Service Group with Solaris zone support

Index