Cluster Server Agent for Oracle Application Server and Oracle Fusion Middleware Installation and Configuration Guide

AIX, Linux, Solaris

6.2

VERITAS™

Cluster Server Agent for Oracle Application Server and Oracle Fusion Middleware Installation and Configuration Guide

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

Agent Version: 6.2

Document version: 6.2 Rev 0

Legal Notice

Symantec Corporation
350 Ellis Street
Mountain View, CA 94043

http://www.symantec.com

Technical Support

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

Symantec's support offerings include the following:

- A range of support options that give you the flexibility to select the right amount of service for any size organization

- Telephone and/or Web-based support that provides rapid response and up-to-the-minute information

- Upgrade assurance that delivers software upgrades

- Global support purchased on a regional business hours or 24 hours a day, 7 days a week basis

- Premium service offerings that include Account Management Services

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

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

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

Contacting Technical Support

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

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

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

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

- Product release level

- Hardware information

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

Licensing and registration

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

www.symantec.com/business/support/

Customer service

Customer service information is available at the following URL:

www.symantec.com/business/support/

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

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

Support agreement resources

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

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

Documentation

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

https://sort.symantec.com/documents

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

doc_feedback@symantec.com

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

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

About Symantec Connect

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

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

Contents

Chapter 3   Installing, upgrading, and removing the agent for Oracle Application Server and Oracle Fusion Middleware 25

Introducing the agent for Oracle Application Server and Oracle Fusion Middleware

This chapter includes the following topics:

- About the Cluster Server agent for Oracle Application Server and Oracle Fusion Middleware

- Supported software

- How the agent makes Oracle Application Server and Oracle Fusion Middleware highly available

- How the agent supports intelligent resource monitoring

- Integrating with the high availability framework of Oracle Application Server

- Oracle Application Server agent functions

- Setting up Oracle Application Server and Oracle Fusion Middleware in a VCS cluster

About the Cluster Server agent for Oracle Application Server and Oracle Fusion Middleware

Cluster Server (VCS) 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 Cluster Server agent for Oracle Application Server, you can ensure application availability. You can also implement disaster recovery using the Global Cluster option.

See the Agent Pack Release Notes for the latest updates or software issues for this agent.

The Cluster Server agent for Oracle Application Server brings the Oracle instances online, monitors the instances, and takes the instances offline. The agent monitors the system processes and server states, and shuts down the server in case of a failover.

The Cluster Server 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.

Supported software

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

How the agent makes Oracle Application Server and Oracle Fusion Middleware highly available

The agent provides the following levels of application monitoring:

- Primary or Basic monitoring
  This mode has Process check and Health check monitoring options. With the
  default Process check option, the agent verifies that the Oracle AS instance
  processes are present in the process table. Process check cannot detect whether
  processes are in the hung or stopped states.

- Secondary or Detail monitoring
  In this mode, the agent runs a utility to verify the status of the Oracle AS instance.
  The agent detects application failure if the monitoring routine reports an improper
  function of the Oracle AS instance processes. When this application failure
  occurs, the Oracle AS instance service group fails over to another node in the
  cluster.
  In addition to these levels of application monitoring, the agent for Oracle
  Application Server and Oracle Fusion Middleware is IMF-aware and uses
  asynchronous monitoring framework (AMF) kernel driver for IMF notification.
  Thus, the agent ensures high availability for Oracle AS instances.

High availability for Oracle Application Server and Oracle Fusion Middleware instances running in Solaris zones

Solaris provides a means of virtualizing operating system services, allowing one or
more processes to run in isolation from other activity on the system. Such a 'sandbox'
is called a 'non-global zone'. Each zone can provide a rich and customized set of
services. The processes that run in a 'global zone' have the same set of privileges
that are available on a Solaris system today.

VCS provides high availability to applications running in non-global zones by extending the failover capability to zones. VCS is installed in a global zone, and all the agents and the engine components run in the global zone. For applications running within non-global zones, agents run script entry points inside the zones. If a zone configured under VCS control faults, VCS fails over the entire service group containing the zone.

For more details, refer to the *Cluster Server Administrator's Guide*.

The Cluster Server agent for Oracle Application Server and Oracle Fusion Middleware is zone-aware and can monitor Oracle Application Server instances running in non-global zones.

How the agent supports intelligent resource monitoring

With Intelligent Monitoring Framework (IMF), VCS supports intelligent resource monitoring in addition to the poll-based monitoring. Poll-based monitoring polls the resources periodically whereas intelligent monitoring performs asynchronous monitoring.

When an IMF-enabled agent starts up, the agent initializes the Asynchronous Monitoring Framework (AMF) kernel driver. After the resource is in a steady state, the agent registers with the AMF kernel driver, the details of the resource that are required to monitor the resource. For example, the agent for Oracle Application Server and Oracle Fusion Middleware registers the PIDs of the Oracle Application Server processes with the AMF kernel driver. The agent's imf_getnotification function waits for any resource state changes. When the AMF kernel driver module notifies the imf_getnotification function about a resource state change, the agent framework runs the monitor agent function to ascertain the state of that resource. The agent notifies the state change to VCS, which then takes appropriate action.

Refer to the *Cluster Server Administrator's Guide* for more information.

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.

See "Configuring the OPMN daemon to work in a cluster" on page 72.

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 Cluster Server agent for Oracle Application Server 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 running the `emctl start` command.

- Starts the OPMN Managed Processes by running 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 running the `emctl stop` command.

- Stops the OPMN Managed Processes by running 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.
The agent for Oracle Application Server and Oracle Fusion Middleware also supports IMF in the first-level check. IMF enables intelligent resource monitoring. The agent for Oracle Application Server and Oracle Fusion Middleware is IMF-aware and uses the AMF kernel driver for resource state change

notifications. See "How the agent supports intelligent resource monitoring" on page 13. You can use the MonitorFreq key of the IMF attribute to specify the frequency at which the agent invokes the monitor function. See "MonitorFreq" on page 60.

■ 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 53.

■ 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 55.

Clean

In case of a failure or after an unsuccessful attempt to bring online or take 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 Oracle Application Server, if the instance does not shut down normally.

Note: For information about the additional functions of the agent for Oracle Application Server and Oracle Fusion Middleware when IMF is enabled: See "Agent functions for the IMF functionality" on page 58.

Setting up Oracle Application Server and Oracle Fusion Middleware in a VCS cluster

Follow the steps below to set up Oracle Application Server and Oracle Fusion Middleware in a cluster:

■ Set up a VCS cluster.
For more information on installing and configuring Cluster Server, refer to the Cluster Server installation and configuration guides.

- Install and configure Oracle Application Server and Oracle Fusion Middleware for High Availability.

  See "About installing Oracle Application Server and Oracle Fusion Middleware for high availability" on page 19.

  See "About configuring the Cluster Server agent for Oracle Application Server and Oracle Fusion Middleware" on page 46.

- Install the Cluster Server agent for Oracle Application Server and Oracle Fusion Middleware.

  See "Installing the agent in a VCS environment" on page 30.

  See "Installing the agent in VCS One environment" on page 33.

- Configure the service groups for Oracle Application Server and Oracle Fusion Middleware.

  See "About configuring service groups for Oracle Application Server and Oracle Fusion Middleware" on page 65.

Installing and configuring Oracle Application Server and Oracle Fusion Middleware for high availability

This chapter includes the following topics:

■

■

■

■

About Oracle Application Server

Oracle Application Server is a standards-based application server that provides a comprehensive and fully integrated platform for running Web sites, J2EE applications, and Web services. It provides full support for the J2EE platform, XML, emerging Web services, and grid standards.

It allows you to save on Web site infrastructure by deploying your fast, scalable Internet applications through built-in Web caching, load balancing and clustering capabilities.

Oracle Application Server provides components that address the development and deployment challenges common to e-business Web sites. These solutions include:

- J2EE and Internet applications

- Portals

- Wireless

- Business Intelligence (Oracle Application Server Forms Services, Report Server, Discoverer.)

- E-business integration

- Web Cache

- System Management (Oracle Enterprise Manager 10g)

- Identity Management and Security

About installing Oracle Application Server and Oracle Fusion Middleware for high availability

The guidelines for installing Oracle Application Server for high availability are as follows:

- Install the Oracle Application Server instance binaries on shared disks.

- When installing Oracle Application Server, ensure that the login_id, id_name, group_id, and group_name for the Oracle Application Server is the same on all the nodes.

- The user and the group must be local and not Network Information Service (NIS and NIS+) users.

The guidelines for installing Oracle Fusion Middleware for high availability are as follows:

- Install the Oracle Fusion Middleware instance binaries on shared disks.

- When installing Oracle Fusion Middleware, ensure that the login_id, id_name, group_id, and group_name for the Oracle Fusion Middleware is the same on all the nodes.

- The user and the group must be local and not Network Information Service (NIS and NIS+) users.

Installing Oracle Application Server for clustering purposes

This section provides the tasks to be performed for installing the Oracle Application Server for clustering purposes:

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.

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 21.

See "Installing the Oracle Application Server software on a shared disk file system" on page 21.

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 recommends 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 the 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 the **Configuration Options** dialog box.

- In the Oracle Universal Installer, select the High Availability or Replication option and select the 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.

Setting up Solaris zones for Oracle Application Server

An example of creating a zone for Oracle Application Server on Solaris is as follows:

Step 1: Create the zone.

```
bash-3.00# zonecfg -z OraAS_zone
OraAS_zone: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:OraAS_zone> create
zonecfg:OraAS_zone> set zonepath=/export/zones/OraAS_zone
```

Step 2: Add the network information to the zone configuration.

```
zonecfg:OraAS_zone> add net
zonecfg:OraAS_zone:net> set address=10.212.98.193
zonecfg:OraAS_zone:net> set physical=bge0
zonecfg:OraAS_zone:net> end
```

Step 3: Add a comment for the zone. This step is optional.

```
zonecfg:OraAS_zone> add attr
zonecfg:OraAS_zone:attr> set name=comment
zonecfg:OraAS_zone:attr> set type=string
zonecfg:OraAS_zone:attr> set value="This is
OraAS_zone zone for Oracle Application Server."
zonecfg:OraAS_zone:attr> end
```

Step 4: Verify and commit the zone configuration.

```
zonecfg:OraAS_zone> verify
zonecfg:OraAS_zone> commit
zonecfg:OraAS_zone> exit

bash-3.00# zoneadm list -cv
ID NAME STATUS PATH
```

```
0 global running /
- OraAS_zone configured /export/zones/OraAS_zone
```

Step 5: Install the zone.

```
bash-3.00# zoneadm list -cv
ID NAME STATUS PATH
0 global running /
- OraAS_zone configured /export/zones/OraAS_zone

bash-3.00# zoneadm -z OraAS_zone install
Preparing to install zone <OraAS_zone>.
Creating list of files to copy from the global zone.
Copying <6208> files to the zone.
Initializing zone product registry.
Determining zone package initialization order.
Preparing to initialize <1420> packages on the zone.
Initialized <1420> packages on zone.
Zone <OraAS_zone> is initialized.
Installation of <113> packages was skipped.
Installation of these packages generated warnings: <VRTSat>
The file </export/zones/OraAS_zone/root/var/sadm/system/logs/install_log>
contains a log ofthe zone installation.

bash-3.00# zoneadm list -cv
ID NAME STATUS PATH
0 global running /
- OraAS_zone installed /export/zones/OraAS_zone
```

Step 6: Configure the zone.

```
To configure the zone for the first time login to the zone console
from the first terminal with the following command:

bash-3.00# zlogin -C OraAS_zone
[Connected to zone 'OraAS_zone' console]
Now, from the second terminal, start the zone.

bash-3.00# zoneadm -z OraAS_zone boot
You will see the following message on the first terminal.
[NOTICE: Zone booting up]
SunOS Release 5.10 Version Generic_118833-36 64-bit
Copyright 1983-2006 Sun Microsystems,Inc. All rights reserved.
Use is subject to license terms.
```

```
Hostname: OraAS_zone
Loading smf(5) service descriptions: 25/25
Select a Language
0. English
1. Japanese
2. Korean
3. Simplified Chinese
4. Traditional Chinese
Please make a choice (0 - 4), or press h or ? for help:
```

For more information on setting up zones, refer to the *Solaris 10 Administration Guide*.

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

This chapter includes the following topics:

- Before you install the Cluster Server agent for Oracle Application Server and Oracle Fusion Middleware

- About the ACC library

- Installing the ACC library

- Installing the agent in a VCS environment

- Installing the agent in VCS One environment

- Uninstalling the agent in a VCS environment

- Removing the agent in VCS One environment

- Removing the ACC library

- Upgrading the agent in a VCS environment

- Upgrading the agent in a VCS One environment

Before you install the Cluster Server agent for Oracle Application Server and Oracle Fusion Middleware

You must install the Cluster Server agent for Oracle Application Server and Oracle Fusion Middleware on all the systems that will host Oracle Application Server service groups.

Before you install the agent for Oracle Application Server and Oracle Fusion Middleware, ensure that the following prerequisites are met.

For VCS, do the following:

■ Install and configure Cluster Server.
  For more information on installing and configuring Cluster Server, refer to the Cluster Server installation and configuration guides.

■ Remove any previous version of this agent.
  To remove the agent,
  See "Uninstalling the agent in a VCS environment" on page 37.

■ Install the latest version of ACC Library.
  To install or update the ACC Library package, locate the library and related documentation in the Agent Pack tarball,
  See "Installing the ACC library" on page 27.

For VCS One, do the following:

■ Install and configure Veritas Cluster Server One.
  For more information on installing and configuring Veritas Cluster Server One, refer to the *Veritas Cluster Server One Installation Guide*.

■ Remove any previous version of this agent.
  To remove the agent,
  See "Removing the agent in VCS One environment" on page 39.

Prerequisites for installing the agent to support Solaris zones

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

■ 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 or 5.1 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 the ACC library

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

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

Note: The LogDbg attribute should be used to enable debug logs for the ACCLib-based agents when the ACCLib version is 6.2.0.0 or later and VCS version is 6.2 or later.

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 a superuser.

2   Download ACC Library.

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

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

AIX	*cd1*/aix/vcs/application/acc_library/*version*_library/pkgs
Linux	*cd1*/linux/generic/vcs/application/acc_library/*version*_library/rpms
Solaris	*cd1*/solaris/*dist_arch*/vcs/application/acc_library/*version*_library/pkgs
	where *dist_arch* is sol_sparc.

4   If you downloaded the individual ACCLib tar file, navigate to the pkgs directory (for AIX and Solaris), or rpms directory (for Linux).

5   Install the package. Enter **Yes**, if asked to confirm overwriting of files in the existing package.

AIX	`# installp -ac -d VRTSacclib.bff VRTSacclib`
Linux	`# rpm -i \` `VRTSacclib-VersionNumber-GA_GENERIC.noarch.rpm`
Solaris	`# pkgadd -d VRTSacclib.pkg`

Note: The LogDbg attribute should be used to enable debug logs for the ACCLib-based agents when the ACCLib version is 6.2.0.0 or later and VCS version is 6.2 or later.

Installing the ACC library IPS package on Oracle Solaris 11 systems

To install the ACC library IPS package on an Oracle Solaris 11 system

1   Copy the `VRTSacclib.p5p` package from the pkgs directory to the system in the `/tmp/install` directory.

2   Disable the publishers that are not reachable as package install may fail, if any, of the already added repositories are unreachable.

```
# pkg set-publisher --disable <publisher name>
```

3   Add a file-based repository in the system.

```
# pkg set-publisher -g /tmp/install/VRTSacclib.p5p Symantec
```

4   Install the package.

```
# pkg install --accept VRTSacclib
```

5   Remove the publisher from the system.

```
# pkg unset-publisher Symantec
```

6   Enable the publishers that were disabled earlier.

```
# pkg set-publisher --enable <publisher name>
```

Installing the ACC library package on Solaris brand non-global zones

With Oracle Solaris 11, you must install the ACC library package inside non-global zones. The native non-global zones are called Solaris brand zones.

To install the ACC library package on Solaris brand non-global zones

1 Ensure that the SMF service
`svc:/application/pkg/system-repository:default` and
`svc:/application/pkg/zones-proxyd:default` are online on the global zone.

```
# svcs svc:/application/pkg/system-repository:default
```

```
# svcs svc:/application/pkg/zones-proxyd:default
```

2 Log on to the non-global zone as a superuser.

3 Ensure that the SMF service
`svc:/application/pkg/zones-proxy-client:default` is online inside the non-global zone:

```
# svcs svc:/application/pkg/zones-proxy-client:default
```

4 Copy the `VRTSacclib.p5p` package from the pkgs directory to the non-global zone (for example, at the `/tmp/install` directory).

5 Disable the publishers that are not reachable, as package install may fail if any of the already added repositories are unreachable.

```
# pkg set-publisher --disable <publisher name>
```

6 Add a file-based repository in the non-global zone.

```
# pkg set-publisher -g/tmp/install/VRTSacclib.p5p Symantec
```

7 Install the package.

```
# pkg install --accept VRTSacclib
```

8 Remove the publisher on the non-global zone.

```
# pkg unset-publisher Symantec
```

9 Clear the state of the SMF service, as setting the file-based repository causes the SMF service `svc:/application/pkg/system-repository:default` to go into the maintenance state.

```
# svcadm clear svc:/application/pkg/system-repository:default
```

10 Enable the publishers that were disabled earlier.

```
# pkg set-publisher --enable <publisher>
```

Note: Perform steps 2 through 10 on each non-global zone.

Installing the agent in a VCS environment

Install the agent for Oracle Application Server and Oracle Fusion Middleware on each node in the cluster.

To install the agent in a VCS environment

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

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

2   Uncompress the file to a temporary location, say /tmp.

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

AIX	`cd1/aix/vcs/application/oracleas_agent/`
	`vcs_version/version_agent/pkgs`
Linux	`cd1/linux/generic/vcs/application/oracleas_agent/`
	`vcs_version/version_agent/rpms`
Solaris	`cd1/solaris/dist_arch/vcs/application/oracleas_agent/`
	`vcs_version/version_agent/pkgs`
	where, *dist_arch* is sol_sparc.

If you downloaded the individual agent tar file, navigate to the pkgs directory (for AIX and Solaris), or rpms directory (for Linux).

4   Log in as a superuser.

5 Install the package.

AIX	`# installp -ac -d` `VRTSvcsoas9.rte.bff VRTSvcsoas9.rte`
Linux	`# rpm -ihv \` `VRTSvcsoas9-AgentVersion-GA_GENERIC.noarch.rpm`
Solaris	`# pkgadd -d . VRTSvcsoas9`

6 After installing the agent package, you must import the agent type configuration file.

Installing the agent IPS package on Oracle Solaris 11 systems

To install the agent IPS package on an Oracle Solaris 11 system

1 Copy the `VRTSvcsoas9.p5p` package from the pkgs directory to the system in the `/tmp/install` directory.

2 Disable the publishers that are not reachable as package install may fail if any of the already added repositories are unreachable.

```
# pkg set-publisher --disable <publisher name>
```

where the publisher name is obtained using the `pkg publisher` command.

3 Add a file-based repository in the system.

```
# pkg set-publisher -g /tmp/install/VRTSvcsoas9.p5p Symantec
```

4 Install the package.

```
# pkg install --accept VRTSvcsoas9
```

5 Remove the publisher from the system.

```
# pkg unset-publisher Symantec
```

6 Enable the publishers that were disabled earlier.

```
# pkg set-publisher --enable <publisher name>
```

Installing agent packages on Solaris brand non-global zones

With Oracle Solaris 11, you must install the agent package inside non-global zones. The native non-global zones are called Solaris brand zones.

To install the agent package on Solaris brand non-global zones

1 Ensure that the SMF service
`svc:/application/pkg/system-repository:default` and
`svc:/application/pkg/zones-proxyd:default` are online on the global
zone.

```
# svcs svc:/application/pkg/system-repository:default
```

```
# svcs svc:/application/pkg/zones-proxyd:default
```

2 Log on to the non-global zone as a superuser.

3 Ensure that the SMF service
`svc:/application/pkg/zones-proxy-client:default` is online inside
non-global zone:

```
# svcs svc:/application/pkg/zones-proxy-client:default
```

4 Copy the `VRTSvcsoas9.p5p` package from the pkgs directory to the non-global
zone (for example, at the `/tmp/install` directory).

5 Disable the publishers that are not reachable, as package install may fail if any
of the already added repositories are unreachable.

```
# pkg set-publisher --disable <publisher name>
```

6 Add a file-based repository in the non-global zone.

```
# pkg set-publisher -g/tmp/install/VRTSvcsoas9.p5p Symantec
```

7 Install the package.

```
# pkg install --accept VRTSvcsoas9
```

8 Remove the publisher on the non-global zone.

```
# pkg unset-publisher Symantec
```

9 Clear the state of the SMF service, as setting the file-based repository causes
the SMF service `svc:/application/pkg/system-repository:default` to go
into the maintenance state.

```
# svcadm clear svc:/application/pkg/system-repository:default
```

10 Enable the publishers that were disabled earlier.

```
# pkg set-publisher --enable <publisher>
```

Note: Perform steps 2 through 10 on each non-global zone.

Installing the agent in a Solaris 10 brand zone

To install the Oracle Application Server and Oracle Fusion Middleware agent in a Solaris 10 brand zone:

- Ensure that the ACC library package, VRTSacclib, is installed in the non-global zone.

  To install VRTSacclib in the non-global zone, run the following command from the global zone:

  ```
  # pkgadd -R /zones/zone1/root -d VRTSacclib.pkg
  ```

- To install the agent package in the non-global zone, run the following command from the global zone:

  ```
  # pkgadd -R zone-root/root -d . VRTSvcsoas9
  ```

  For example: `# pkgadd -R /zones/zone1/root -d . VRTSvcsoas9`

Installing the agent in VCS One environment

You must install the agent for Oracle Application Server on all the client systems of the VCS One cluster that will host the Oracle Application Server service group. You can install the agent for Oracle Application Server using the installagpack program or using the command line interface (CLI).

The installation of the agent packs involves the following phases:

Installing the agent packages	See "Installing the agent packages using the installer" on page 33.
Adding the agent resource type definitions	See "Adding the agent resource type definitions to the Policy Master Server on UNIX" on page 35.
	See "Adding the agent resource type definitions to the Policy Master Server on Windows" on page 37.

Note: The installagpack program supports only the -addtypes, -rmtypes, -responsefile, and -rsh options. Symantec recommends that you do not use any of the other options from the installagpack command help output.

Installing the agent packages using the installer

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

> **Note:** To install the VCS One client for managing VMware ESX Servers, download the tar ball for Red Hat Enterprise Linux 4 (RHEL 4) x86 (32-bit) or RHEL 5 x86_64

Perform the following steps to install the agent packages using the installer

1   On the Policy Master system, download the complete Agent Pack tarball or the individual agent tarball from the Symantec Operations Readiness Tools (SORT) site: https://sort.symantec.com/agents.

2   Uncompress the file to a temporary location, say /tmp.

3   If you downloaded the complete Agent Pack tarball, navigate to the following directory containing the installer for the VCS One agents, for the platform running in your environment:

AIX	`cd1/aix/vcsone/`*vcsone_version*
Linux	`cd1/linux/`*dist_arch*`/vcsone/`*vcsone_version*
	Where *dist* is the Linux distribution and *arch* is the architecture.
Solaris	`cd1/solaris/`*dist_arch*`/vcsone/`*vcsone_version*
	Where, *dist_arch* is 'sol_sparc'.

4   Enter the following command to start the agent pack installation:

    # ./installagpack [-rsh]

You can use the `-rsh` option if rsh and rcp are used for communication between systems instead of the default ssh and scp. This option requires that systems be preconfigured such that the `rsh` commands between systems execute without prompting for passwords or confirmations.

5   Enter the name of the client systems where you want to install the agents.

6   Choose whether to install all the agents or any specific agent. Follow the installer prompt to specify your option.

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

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

Installing the agent package using the CLI

You can install the desired agent package using the CLI, on one or more client systems of a specific platform type.

Perform the following steps to install the agent packages using CLI

1 On the Policy Master system, download the complete Agent Pack tarball or the individual agent tarball from the Symantec Operations Readiness Tools (SORT) site: https://sort.symantec.com/agents.

2 Uncompress the file to a temporary location, say /tmp.

3 If you downloaded the complete Agent Pack tarball, navigate to the following directory containing the installer for the VCS One agents, for the platform running in your environment:

AIX	# **cd1/aix/vcsone/vcsone_version/pkgs**
Linux	# **cd1/linux/*dist_arch*/vcsone/vcsone_version/rpms**
	Where, *dist* is the Linux distribution and *arch* is the architecture
Solaris	# **cd1/solaris/*dist_arch*/vcsone/vcsone_version/pkgs**
	Where *dist_arch* is 'sol_sparc'.

4 Type the following command on each client system to install the agent. Answer the prompt accordingly:

AIX	# **installp -ac -d . VRTSvcsoas9.rte**
Linux	# **rpm -ivh VRTSvcsoas9_rpm_filename**
Solaris	For Solaris 10:
	# **pkgadd -d . VRTSvcsoas9**
	For Solaris 11: See "Installing the agent IPS package on Oracle Solaris 11 systems" on page 31.

Adding the agent resource type definitions to the Policy Master Server on UNIX

You must add the agent resource type definitions to the Policy Master database configuration. You can perform this task from any client system in the VCS One cluster.

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

To add the agent resource types to the policy master database configuration

1   Set up RSH or SSH communications between the client system and the policy master system.

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

2   Make sure that the PM daemon is running.

```
# /opt/VRTSvcsone/bin/haclus -display
```

The output should show ClusterState is RUNNING.

3   Access the temporary location where you downloaded the tar ball and depending on the platform type, navigate to the directory containing the agent installer:

AIX	`cd1/aix/vcsone/`*`vcsone_version`*
Linux	`cd1/linux/`*`dist_arch`*`/vcsone/`*`vcsone_version`*
	Where *dist* is the Linux distribution and *arch* is the architecture.
Solaris	`cd1/solaris/`*`dist_arch`*`/vcsone/`*`vcsone_version`*
	Where *dist_arch* is the sol_sparc.

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

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

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

6   Review the output as the installer verifies communication with the Policy Master system.

7   Choose whether to add the type definitions for all the agents or for specific agents. Follow the installer prompts to add the type definitions.

8   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.

Adding the agent resource type definitions to the Policy Master Server on Windows

After you have installed the agent package, you must add the agent resource type definitions to the Policy Master database configuration. You must perform this task from the Policy Master Server.

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

To add the agent resource types to the Policy Master Server on Windows

Perform the following steps from the Policy Master Server command prompt.

1 Create a temporary directory on the Policy Master Server, to add the type definitions.

```
C:\> mkdir addtypes_tmp
```

2 Change your working directory to the temporary directory created in step 1.

```
C:\> chdir addtypes_tmp
```

3 Copy the agent's type xml file in the temporary directory.

4 Convert this type xml file into the type cmd file.

```
C:\addtypes_tmp> haconf -xmltocmd type_xml_filename.xml
```

5 Rename the *type_xml_filename*.xml.cmd file to *type_xml_filename*.bat

6 Run the batch file.

```
C:\addtypes_tmp> type_xml_filename.bat >log.txt 2>&1
```

7 Review the log.txt file for any errors.

8 Verify that the type has been successfully added to the Policy Master Server.

```
C:\addtypes_tmp> hatype -list -platform platform_name
```

Uninstalling the agent in a VCS environment

You must uninstall the agent for Oracle Application Server and Oracle Fusion Middleware 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 running the following command from any node in the cluster:

```
# haconf -makerw
```

3 Remove all Oracle Application Server resources from the cluster. Run 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 running 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 running 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 and Oracle Fusion Middleware from each node in the cluster.

Run the following command to uninstall the agent:

AIX	`# installp -u VRTSvcsoas9.rte`
Linux	`# rpm -e VRTSvcsoas9`
Solaris	`# pkgrm VRTSvcsoas9`
	Note: To uninstall the agent IPS package on a Solaris 11 system, run the following command:
	`# pkg uninstall VRTSvcsoas9`

Removing the agent in VCS One environment

Removing the agent package involves removing the agent files from each client system where it was installed.

You can remove the packages using the agent pack installer or the command line.

See "Removing the agent packages using the installer" on page 39.

See "Removing the agent package using CLI" on page 40.

After removing the agent packages you can remove the agent type definition from the Policy Master system.

See "Removing the agent type definition from the Policy Master system on UNIX" on page 41.

See "Removing the agent type definition from the Policy Master system on Windows" on page 42.

Removing the agent packages using the installer

You can remove all the agent packages or the desired agent package using the uninstallagpack program.

Note: The uninstallagpack program supports only the -responsefile and -rsh options. Symantec recommends that you do not use any of the other options from the uninstallagpack command help output.

To remove the agent packages from the client systems

1 Freeze the service groups that hosts the application, on the system from which you want to remove the agent package.

    # hagrp -freeze <*groupname*>

2 Stop the agent on all client systems before you remove the agent package from the system.

    # haagent -stop -notransition <*AgentName*> -sys <*system_name*>

3 Ensure that the agent operations are stopped on all the cluster systems.

    # haagent -display <*AgentName*>

4 Access the temporary location where you downloaded the Agent Pack and navigate to the directory containing the package for the platform running in your environment:

AIX   **cd1/aix/vcsone/*vcsone_version***

Linux   **cd1/linux/*dist_arch*/vcsone/*vcsone_version***

                  Where *dist* is the Linux distribution and *arch* is the architecture.

Solaris   **cd1/solaris/*dist_arch*/vcsone/*vcsone_version***

                  Where *dist_arch* is the sol_sparc.

5 Start the uninstallagpack program.

   # **./uninstallagpack [-rsh]**

6 Enter the name of the client systems on which you want to uninstall the agent pack. The names must be separated by spaces.

7 Choose whether to remove all the agent packages or a specific agent package. Follow the installer prompt to remove the agent package.

8 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 agent package using CLI

You can remove a desired agent package using the CLI.

Note: You must remove this agent package from each client system in the cluster.

To remove the agent for Oracle Application Server from a client system

◆   Run the following command on each client system to remove the agent. Answer
    prompts accordingly:

AIX	# **installp -u VRTSvcsoas9**
Linux	# **rpm -e VRTSvcsoas9**
Solaris	For Solaris 10:
	# **pkgrm VRTSvcsoas9**
	For Solaris 11:
	# **pkg uninstall VRTSvcsoas9**

Removing the agent type definition from the Policy Master system on UNIX

After you remove the agent packages, you can remove the agent type definitions
for agents you removed, from the Policy Master system.

To remove the agent type definition from the Policy Master system on UNIX

1   Navigate to the following directory on the client system.

    # **cd /opt/VRTS/install**

2   Run the following command to remove the agent type definition from the Policy
    Master system:

    # **./installagpack -rmtypes**

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

4   Choose whether to remove the type definitions for all the agents or for specific
    agents. Follow the installer prompts to remove the type definitions.

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

Removing the agent type definition from the Policy Master system on Windows

After you remove the agent packages, you can remove the agent type definitions for agents you removed, from the Policy Master system.

To remove the agent type definition from the Policy Master system on Windows

◆ Run the following command from the Policy Master Server command prompt.

```
C:\> hatype -delete agentname_i.e._typename -platform platformname
```

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`
Linux	`# rpm -e VRTSacclib`
Solaris	`# pkgrm VRTSacclib`
	Note: To uninstall the ACCLib IPS package on a Solaris 11 system, run the following command:
	`# pkg uninstall VRTSacclib`

Upgrading the agent in a VCS environment

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

To upgrade the agent in a VCS environment

1 Persistently freeze the service groups that host the application.

```
# hagrp -freeze GroupName -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. Use the platform's native software management program to remove the agent for Oracle Application Server and Oracle Fusion Middleware from each node in the cluster.

Run the following command to uninstall the agent:

AIX	# installp -u VRTSvcsoas9.rte
Linux	# rpm -e VRTSvcsoas9
Solaris	For Solaris 10:
	# pkgrm VRTSvcsoas9
	For Solaris 11:
	# **pkg uninstall VRTSvcsoas9**

5 Install the new agent on all the nodes.

See "Installing the agent in a VCS environment" on page 30.

6 Copy the new `OracleAS9Types.cf` file from the agent's conf directory, to the VCS conf directory `/etc/VRTSvcs/conf/config`.

VCS 4.x	■ AIX ■ Linux ■ Solaris	/etc/VRTSvcs/conf/sample_OracleAS9/ OracleAS9Types.cf
VCS 5.x or later	■ AIX ■ Linux	/etc/VRTSagents/ha/conf/OracleAS9/ OracleAS9Types.cf
VCS 5.0	■ Solaris SPARC	/etc/VRTSagents/ha/conf/OracleAS9/ OracleAS9Types50.cf
VCS 5.1 or later	■ Solaris SPARC	/etc/VRTSagents/ha/conf/OracleAS9/ OracleAS9Types51.cf

Note: If you are using Solaris SPARC, copy the `OracleAS9Types50.cf` file for VCS 5.0 (and its intermediate Maintenance Packs) and `OracleAS9Types51.cf` file for VCS 5.1 or later.

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 SystemName
```

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

```
# hagrp -unfreeze GroupName -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.

To upgrade the agent with minimal disruption, in a VCS One environment

1 Freeze service groups that hosts the application.

```
# hagrp -freeze -propagate GroupName
```

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

```
# hastop -client -sys SystemName -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. Type the following command on each client system to remove the agent. Answer prompts accordingly:

AIX   `# installp -u VRTSvcsoas9`

Linux   `# rpm -e VRTSvcsoas9`

Solaris   For Solaris 10:

                            `# pkgrm VRTSvcsoas9`

                            For Solaris 11:

                            `# pkg uninstall VRTSvcsoas9`

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

See "Installing the agent in VCS One environment" on page 33.

6 Add the agent types, using the installagpack program.

See "Adding the agent resource type definitions to the Policy Master Server on UNIX" on page 35.

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 SystemName
```

10 Unfreeze the service groups.

```
# hagrp -unfreeze -propagate GroupName
```

Configuring the agent for Oracle Application Server and Oracle Fusion Middleware

This chapter includes the following topics:

About configuring the Cluster Server agent for Oracle Application Server and Oracle Fusion Middleware

After installing the Cluster Server agent for Oracle Application Server and Oracle Fusion Middleware, you must import the agent type configuration file. After importing this file, review the attributes table that describes the resource type and its attributes, and then create and configure Oracle Application Server resources.

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

See "About sample configurations for the agents for Oracle Application Server and Oracle Fusion Middleware" on page 82.

Importing the agent types files in a VCS environment

To use the agent for Oracle Application Server and Oracle Fusion Middleware, you must import the agent types file into the cluster.

You can import the agent types file using the VCS graphical user interface or using the command line interface.

To import the agent types file using the VCS graphical user interface

1   Start the Cluster Manager (Java Console) 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:

   ■   For VCS 5.x and later on AIX and Linux

      `/etc/VRTSagents/ha/conf/OracleAS9/OracleAS9Types.cf`

   ■   For VCS 5.1 and later on Solaris SPARC

      `/etc/VRTSagents/ha/conf/OracleAS9/OracleAS9Types51.cf`

   ■   For VCS 5.0 on Solaris SPARC

      `/etc/VRTSagents/ha/conf/OracleAS9/OracleAS9Types50.cf`

   ■   For VCS 4.x on AIX, Linux, and Solaris

      `/etc/VRTSvcs/conf/sample_OracleAS9/OracleAS9Types.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 *Cluster Server Administrator's Guide*.

To import the agent types file using the command line interface (CLI):

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

2 Run the following command:

```
#sh /etc/VRTSagents/ha/conf/OracleAS9types.cmd
```

3 Run the following command to verify that the agent types file is successfully imported to the VCS engine:

```
#hatype -display OracleS9
```

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

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

Oracle Application Server agent attributes

Table 4-1 lists the required attributes.

Table 4-1 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.
	For details on the values of this attribute for different components, See Table 4-2 on page 50.
	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 4-1   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
	Note: The use of the ResLogLevel attribute is deprecated from VCS version 6.2 onwards. You must use the LogDbg attribute instead of the ResLogLevel attribute to enable debug logs for the ACCLib-based agents, when the ACCLib version is 6.2.0.0 or later. The agent captures the first failure data of the unexpected events and automatically logs debug messages in their respective agent log files.

Table 4-1   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 4-2 lists the values of the ComponentProcessType attribute.

Table 4-2   ComponentProcessType attribute values

Component	Values of ComponentProcessType attribute
Oracle Infrastructure components	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. ■ OVD, for OVD (Oracle Virtual Directory) component type.

Table 4-2   ComponentProcessType attribute values *(continued)*

Component	Values of ComponentProcessType attribute
Oracle Application Server components	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, performance_server, messaging_server for wireless component type. ■ ServicesStatus, PreferenceServer, SessionServer, OAD, OSAgent, for Discoverer component type. **Note:** Check the output of the 'opmnctl status' command to know the configured process types for Discoverer component type.

Enterprise Manager components	The attribute values for Enterprise Manager components are:
	
	■ iasconsole, for iasconsole component type.
	■ dbconsole, for dbconsole component type.
	■ agent, for EMagent component type.
	■ OHS, for Oracle HTTP server.
	    This attribute value is only applicable to Enterprise Manager Grid Control 11g and 12c.
	■ ReportsServerComponent, for report server.
	    This attribute value is applicable to Oracle Application Server 10g and Oracle Fusion Middleware.

Oracle Application Server SOA suite components	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.

Table 4-3 lists the optional attributes

Table 4-3 Optional attributes

Attribute	Description
LogDbg	For ACCLib-based agents, you must use the LogDbg resource type attribute to enable the debug logs when the ACCLib version is 6.2.0.0 or later and the VCS version is 6.2 or later.
	Set the LogDbg attribute to DBG_5 to enable debug logs for ACCLIB based agent. By default, setting the LogDbg attribute to DBG_5 enables debug logs for all OracleAS9 resources in the cluster. If debug logs must be enabled for a specific OracleAS9 resource, override the LogDbg attribute.
	Type and dimension: string-keylist
	Default: {} (none)
	For more information on how to use the LogDbg attribute, refer to the *Cluster Server Administrator's Guide*.
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 55.
	Type and dimension: string-scalar
	Default: ""
	Example: `/u01/app/oracle/Infra/opmn/bin/oasmonitor.sh`
OracleInstanceHome	Absolute path name of the instance directory.
	Note: Specify this instance for Oracle Fusion Middleware 11g only.
	Default: ""
	Example: `/home/oracle/Oracle/Middleware/asinst_1`
OracleInstanceName	Name of the Oracle instance configured for the particular installation.
	Note: Specify this instance for Oracle Fusion Middleware 11g only.
	Default: ""
	Example: asinst_1

Table 4-3   Optional attributes *(continued)*

Attribute	Description
SecondLevelMonitor	Used to enable second-level monitoring. Second-level monitoring (SLM) 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 53.
	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

Note: For information about the additional attributes of the agent for Oracle Application Server and Oracle Fusion Middleware when IMF is enabled: See "Attributes that enable IMF" on page 59.

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 execute 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 execute the `emctl status <iasconsole/dbconsole>` command and check 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 an 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 is to be installed, 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:

See "Sample service group configurations for Solaris zone support" on page 100.

Enabling the agent for Oracle Application Server and Oracle Fusion Middleware to support IMF

This chapter includes the following topics:

- About Intelligent Monitoring Framework
- Agent functions for the IMF functionality
- Attributes that enable IMF
- Before you enable the agent to support IMF
- Enabling the agent to support IMF
- Disabling intelligent resource monitoring

About Intelligent Monitoring Framework

With the IMF feature, VCS supports intelligent resource monitoring in addition to the poll-based monitoring. Poll-based monitoring polls the resources periodically whereas intelligent monitoring performs asynchronous monitoring. You can enable or disable the intelligent resource monitoring functionality of the Oracle Application Server and Oracle Fusion Middleware agent.

VCS process and mount-based agents use the AMF kernel driver that provides asynchronous event notifications to the agents that are enabled for IMF.

You can enable the Oracle Application Server and Oracle Fusion Middleware agent for IMF, provided the following software versions are installed:

- Cluster Server (VCS) 5.1 SP1 or later

- Cluster Server agent for Oracle Application Server and Oracle Fusion Middleware version 5.1.5.0 or later

Refer to the *Cluster Server Administrator's Guide* for more information about IMF notification module functions and administering the AMF kernel driver.

Benefits of IMF

IMF offers the following benefits:

- Performance
  Enhances performance by reducing the monitoring of each resource at a default of 60 seconds for online resources, and 300 seconds for offline resources. IMF enables the agent to monitor a large number of resources with a minimal effect on performance.

- Faster detection
  Asynchronous notifications would detect a change in the resource state as soon as it happens. Immediate notification enables the agent to take action at the time of the event.

Agent functions for the IMF functionality

If the Oracle Application Server and Oracle Fusion Middleware is enabled for IMF support, the agent supports the following functions, in addition to the functions mentioned in Oracle Application Server agent functions.

imf_init

This function initializes the Oracle Application Server and Oracle Fusion Middleware agent to interface with the AMF kernel driver, which is the IMF notification module for the agent for Oracle Application Server and Oracle Fusion Middleware. This function runs when the agent starts up.

imf_getnotification

This function gets notifications about resource state changes. This function runs after the agent initializes with the AMF kernel module. This function continuously waits for notification and takes action on the resource upon notification.

imf_register

This function registers or unregisters resource entities with the AMF kernel module. This function runs for each resource after the resource goes into a steady state—online or offline.

Attributes that enable IMF

If the agent for Oracle Application Server and Oracle Fusion Middleware is enabled for IMF support, the agent uses the following type-level attributes in addition to the attributes described in Oracle Application Server agent attributes.

IMF

This resource type-level attribute determines whether the Oracle Application Server and Oracle Fusion Middleware agent must perform intelligent resource monitoring. You can also override the value of this attribute at the resource level.

This attribute includes the following keys:

Mode

Define this attribute to enable or disable intelligent resource monitoring. Valid values are as follows:

- 0—Does not perform intelligent resource monitoring

- 1—Performs intelligent resource monitoring for offline resources and performs poll-based monitoring for online resources

- 2—Performs intelligent resource monitoring for online resources and performs poll-based monitoring for offline resources

- 3—Performs intelligent resource monitoring for both online and for offline resources.

Note: The agent for Oracle Application Server and Oracle Fusion Middleware supports intelligent resource monitoring for online resources only. Hence, Mode should be set to either 0 or 2.

Type and dimension: integer-association

Default: 0 for VCS 5.1 SP1, 3 for VCS 6.0 and later.

MonitorFreq

This key value specifies the frequency at which the agent invokes the monitor agent function. The value of this key is an integer.

Default: 1

You can set this key to a non-zero value for cases where the agent requires to perform both poll-based and intelligent resource monitoring.

If the value is 0, the agent does not perform poll-based process check monitoring.

After the resource registers with the AMF kernel driver, the agent calls the monitor agent function as follows:

- After every (MonitorFreq x MonitorInterval) number of seconds for online resources
- After every (MonitorFreq x OfflineMonitorInterval) number of seconds for offline resources

RegisterRetryLimit

If you enable intelligent resource monitoring, the agent invokes the imf_register agent function to register the resource with the AMF kernel driver.

The value of the RegisterRetryLimit key determines the number of times the agent must retry registration for a resource. If the agent cannot register the resource within the limit that is specified, then intelligent monitoring is disabled until the resource state changes or the value of the Mode key changes.

Default: 3.

IMFRegList

An ordered list of attributes whose values are registered with the IMF notification module.

Type and dimension: string-vector

Default: No default value

Note: The attribute values can be overriden at the resource level.

Before you enable the agent to support IMF

Before you enable the Oracle Application Server and Oracle Fusion Middleware agent to support IMF, ensure that the AMF kernel module is loaded and AMF is

configured. For details, refer to the 'Administering the AMF kernel driver' section of the *Cluster Server Administrator's Guide*. For details about the commands you can configure AMF using the `amfconfig -h` command.

Enabling the agent to support IMF

In order to enable the Oracle Application Server and Oracle Fusion Middleware agent to support IMF, you must make the following configuration changes to the attributes of the agent:

■ AgentFile: Set the AgentFile attribute to **Script51Agent**

■ IMF Mode: Set the IMF Mode attribute to **2**

■ IMFRegList: Update the IMFRegList attribute

The following sections provide more information about the commands you can use to make these configuration changes, depending on whether VCS is in a running state or not.

Note: If you have upgraded VCS from an earlier version to version 5.1 SP1 or later, and you already have Oracle Application Server and Oracle Fusion Middleware agent 5.1.5.0 installed, ensure that you run the following commands to create appropriate symbolic links:

```
# cd /opt/VRTSagents/ha/bin/OracleAS9

# ln -s /opt/VRTSamf/imf/imf_getnotification imf_getnotification

# ln -s /opt/VRTSagents/ha/bin/OracleAS9/monitor imf_register
```

If VCS is in a running state

To enable the Oracle Application Server and Oracle Fusion Middleware resource for IMF when VCS is in a running state:

1   Make the VCS configuration writable.

```
# haconf -makerw
```

2   Run the following command to update the AgentFile attribute.

```
# hatype -modify OracleAS9 AgentFile\
/opt/VRTSvcs/bin/Script51Agent
```

3 For VCS version 6.0 or later, run the following commands to add the IMF attributes:

```
# haattr -add -static  OracleAS9 IMF -integer -assoc Mode 0 \
MonitorFreq 1 RegisterRetryLimit 3

# haattr -add -static OracleAS9 IMFRegList -string -vector
```

Note: Run these commands only once after you first enable IMF support for the agent.

4 Run the following command to update the IMF attribute.

```
# hatype -modify OracleAS9 IMF Mode num MonitorFreq num
RegisterRetryLimit num
```

For example, to enable intelligent monitoring of online resources, with the MonitorFreq key set to 5, and the RegisterRetryLimit key is set to 3, run the following command:

```
# hatype -modify OracleAS9 IMF Mode 2 MonitorFreq 5 \
RegisterRetryLimit 3
```

Note: The valid values for the Mode key of the IMF attribute are 0 (disabled) and 2 (online monitoring).

5 Run the following command to update the IMFRegList attribute:

```
# hatype -modify OracleAS9 IMFRegList User OracleHome \
ComponentProcessType
```

6 Save the VCS configuration.

```
# haconf -dump -makero
```

7 If the Oracle Application Server and Oracle Fusion Middleware agent is running, restart the agent.

For information on the commands you can use to restart the agent, see Restarting the agent.

Restarting the agent

To restart the agent:

1 Run the following command to stop the agent forcefully:

```
# haagent -stop OracleAS9 -force -sys <system>
```

Note: Stopping the agent forcefully eliminates the need to take the resource offline.

2 Run the following command to start the agent:

```
# haagent -start OracleAS9 -sys <system>.
```

If VCS is not in a running state

To change the OracleAS9 type definition file when VCS is not in a running state:

1 Update the AgentFile attribute.

```
static str AgentFile = "/opt/VRTSvcs/bin/Script51Agent"
```

2 Update the IMF attribute.

The valid values for the Mode key of the IMF attribute are 0 (disabled) and 2 (online monitoring).

```
static int IMF{} = { Mode=num, MonitorFreq=num,
RegisterRetryLimit=num }
```

For example, to update the IMF attribute such that the Mode key is set to 2, the MonitorFreq key is set to 5, and the RegisterRetryLimit key is set to 3:

```
static int IMF{} = { Mode=2, MonitorFreq=5, RegisterRetryLimit=3
}
```

3 Update the IMFRegList attribute.

```
static str IMFRegList[] = { User, OracleHome, ComponentProcessType
}
```

Disabling intelligent resource monitoring

To disable intelligent resource monitoring

1   Make the VCS configuration writable.

    ```
    # haconf -makerw
    ```

2   To disable intelligent resource monitoring for all the resources of a certain type, run the following command:

    ```
    # hatype -modify OracleAS9 IMF -update Mode 0
    ```

3   To disable intelligent resource monitoring for a specific resource, run the following command:

    ```
    # hares -override resource_name IMF
    ```

    ```
    # hares -modify resource_name IMF -update Mode 0
    ```

4   Save the VCS configuration.

    ```
    # haconf -dump -makero
    ```

Configuring the service groups for Oracle Application Server and Oracle Fusion Middleware using the CLI

This chapter includes the following topics:

- About configuring service groups for Oracle Application Server and Oracle Fusion Middleware

- Before configuring the service groups for Oracle Application Server and Oracle Fusion Middleware

- Configuring an Oracle Application Server in a clustered environment

About configuring service groups for Oracle Application Server and Oracle Fusion Middleware

Configuring the Oracle Application Server and Oracle Fusion Middleware service group involves creating the Oracle Application Server service group, its resources, and defining attribute values for the configured resources. You must have administrator privileges to create and configure a service group.

You can configure the service groups using one of the following:

- The Cluster Manager (Java console)

- Veritas Infoscale Operations Manager

- The command line

Before configuring the service groups for Oracle Application Server and Oracle Fusion Middleware

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

- Verify that Cluster Server is installed and configured on all nodes in the cluster where you will configure the service group.
  For more information on installing and configuring Cluster Server, refer to the Cluster Server installation and configuration guides.

- Verify that the Cluster Server agent for Oracle Application Server and Oracle Fusion Middleware is installed on all nodes in the cluster.
  See "Installing the agent in a VCS environment" on page 30.

Configuring an Oracle Application Server in a clustered environment

The Cluster Server 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

- Obtaining dedicated virtual IP address and DNS name

- Obtaining dedicated user accounts

- Creating service group and supporting resources

- Installing the Oracle Application Server software

- Configuring virtual hostname on a standalone or clustered Oracle Application Server

- Configuring the OPMN daemon to work in a cluster

- Placing the Oracle Application Server component under cluster control

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 48.

For a sample service group configurations:

See "Sample service group configuration for Oracle Application Server" on page 83.

Troubleshooting the agent for Oracle Application Server and Oracle Fusion Middleware

This chapter includes the following topics:

- Using the 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

- Troubleshooting the configuration for IMF

Using the correct software and operating system versions

Ensure that you use correct software and operating system versions.

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

Meeting prerequisites

Before installing the agent for Oracle Application Server and Oracle Fusion Middleware, ensure that the following prerequisites are met.

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

See "Before you install the Cluster Server agent for Oracle Application Server and Oracle Fusion Middleware" on page 26.

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 Cluster Server agent for Oracle Application Server and Oracle Fusion Middleware" on page 26.

- Importing the types.cf file for Solaris zone support
  See "Importing the agent types files in a VCS environment" on page 47.

- Configuring the Oracle Application Server resources for Solaris zone support
  See "Configuring the Oracle Application Server resources for Solaris zone support" on page 55.

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 described 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 the following 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 and Oracle Fusion Middleware, 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 and Oracle Fusion Middleware, you can also access the engine log file for more information about a particular resource.

The VCS engine log file is at `/var/VRTSvcs/log/engine_A.log`.

The VCS One engine log file is at `/var/VRTSvcsone/log/engine_A.log`.

The VCS One client log file is at `/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.

The LogDbg attribute should be used to enable the debug logs for the ACCLib-based agents when the ACCLIB version is 6.2.0.0 or later and the VCS version is 6.2 or later.

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 Save the configuration changes.

```
# haconf -dump
```

7 Review the contents of the log file.

You can also contact Symantec support for more help.

To enable debug logs for all resources of type OracleAS9

◆ Enable the debug log.

```
# hatype -modify OracleAS9 LogDbg DBG_5
```

To override the LogDbg attribute at resource level

◆ Override the LogDbg attribute at the resource level and enable the debug logs for the specific resource.

```
# hares -override OracleAS9 LogDbg
# hares -modify OracleAS9 LogDbg DBG_5
```

Troubleshooting the configuration for IMF

If you face problems with the IMF configuration or functionality, consider the following:

- Ensure that the following attributes are configured with appropriate values.

  - AgentFile

  - IMF

  - IMFRegList
    If IMFRegList is not configured correctly, the Oracle Application Server and Oracle Fusion Middleware resources that have been registered for IMF get unregistered every time the monitor function is run.

- If you have configured the required attributes to enable the Oracle Application Server and Oracle Fusion Middleware agent for IMF, but the agent is still not IMF-enabled, restart the agent. The imf_init function runs only when the agent starts up, so when you restart the agent, imf_init runs and initializes the Oracle Application Server and Oracle Fusion Middleware agent to interface with the AMF kernel driver.

- You can run the following command to check the value of the MonitorMethod attribute and to verify that a resource is registered for IMF.
  ```
  # hares -value resource MonitorMethod system
  ```

The MonitorMethod attribute specifies the monitoring method that the agent uses to monitor the resource:

- Traditional—Poll-based resource monitoring

- IMF—Intelligent resource monitoring

- You can use the `amfstat` to see a list of registered PIDs for a OracleAS9 resource.

  The `amfstat` command shows the PIDs monitored by the Oracle Application Server and Oracle Fusion Middleware agent.

```
[root@vcslx202 OracleAS9]# amfstat
AMF Status Report

Registered Reapers (2):
======================
RID     PID       EVENT   REAPER
3       11436     0       0       InfoSphere
4       24416     4       0       OracleAS9


Process ONLINE Monitors (4):
============================
RID     R_RID   PID       GROUP
28      4       23573     opmn
29      4       23574     opmn
30      4       24206     HTTP_Server
31      4       24717     home
```

The output of the `ps -ef` command:

```
[root@vcslx202 OracleAS9]# ps -ef | grep opmn
oracle  23573     1  0 15:34 ?  00:00:00 /oraas/oracle/Middleware
                                 /OraHome_1/opmn/bin/opmn -d
oracle  23574 23573  0 15:34 ?  00:00:00 /oraas/oracle/Middleware
                                 /OraHome_1/opmn/bin/opmn -d
```

- Run the following command to set the ResLogLevel attribute to TRACE. When you set ResLogLevel to TRACE, the agent logs messages in the OracleAS9_A.log file.

  ```
  # hares -modify ResourceName ResLogLevel TRACE
  ```

  For more information about the ResLogLevel attribute, See "Oracle Application Server agent attributes" on page 48.

- Run the following command to view the content of the AMF in-memory trace buffer.

  ```
  # amfconfig -p dbglog
  ```

Known issues

This release of the agent for Oracle Application Server and Oracle Fusion Middleware has the following known issues:

Problem

An error message might appear when you run the `hares -offline` command to take a resource offline.

Description

When a resource is taken offline, it is unregistered from the AMF module. However, the imf_register function attempts to unregister the resource again.

This results in an error message from the engine log.

Workaround

It is safe to ignore this error message.

Sample Configurations

This appendix includes the following topics:

■ About sample configurations for the agents for Oracle Application Server and Oracle Fusion Middleware

■ Sample agent type definition

■ Sample service group configuration for Oracle Application Server

■ Sample service group configuration for Oracle Fusion Middleware

■ Sample Oracle Application Server resource configuration

■ Sample Oracle Application Server resource configuration for Oracle Fusion Middleware installation

■ Sample IMF configuration

■ Sample service group configurations for Solaris zone support

About sample configurations for the agents for Oracle Application Server and Oracle Fusion Middleware

The sample configuration graphically depicts the resource types, resources, and resource dependencies within the service group. Review these dependencies carefully before configuring the agents for Oracle Application Server and Oracle Fusion Middleware. For more information about these resource types, refer to the *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 AgentDirectory = "/opt/VRTSagents/ha/bin/OracleAS9"
 static str AgentFile = "/opt/VRTSvcs/bin/Script50Agent"
 static int LevelTwoMonitorFreq = 1
 static str ArgList[] = { ResLogLevel, State, IState, User, OracleHome,
                          ComponentProcessType, SecondLevelMonitor,
                          MonitorProgram, OracleInstanceName,
                          OracleInstanceHome }
 static boolean AEPTimeout = 1
 str ResLogLevel = INFO
 str User
 str OracleHome
 str ComponentProcessType
 int SecondLevelMonitor
 str MonitorProgram
 str OracleInstanceName
 str OracleInstanceHome
)
```

Sample service group configuration for Oracle Application Server

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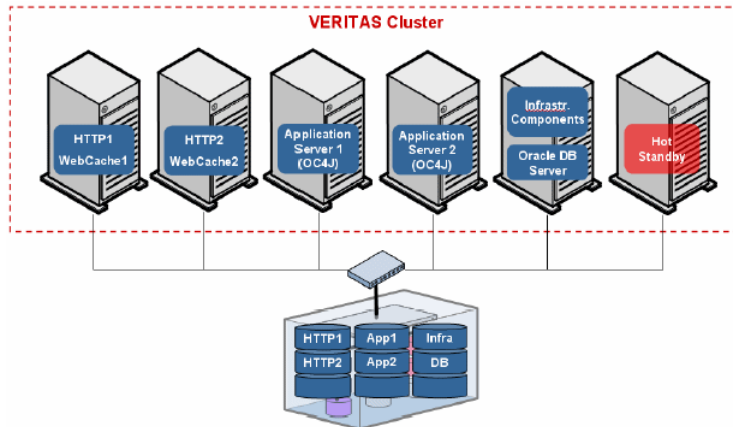
Each application service is installed on shared disk file systems provided by a SAN, which is essential to enable each application service to run on any node in the cluster.

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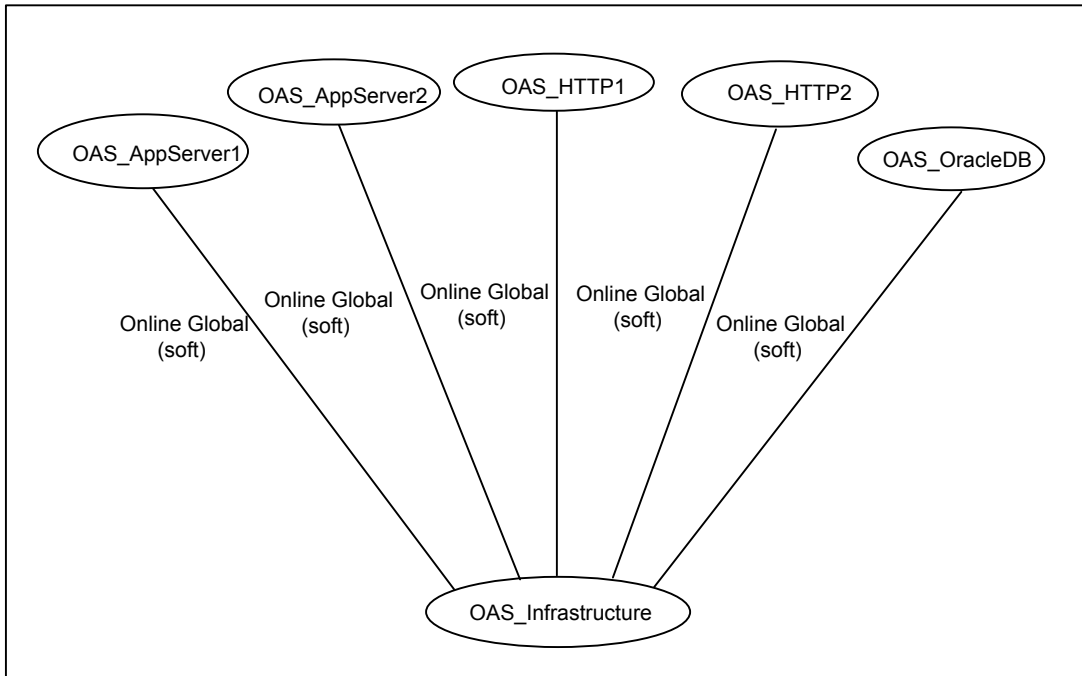
Figure A-3 shows a service group that includes the Oracle database and the Oracle Listener. This group represents the standard production database service that is part of the classic three-tier application environment.

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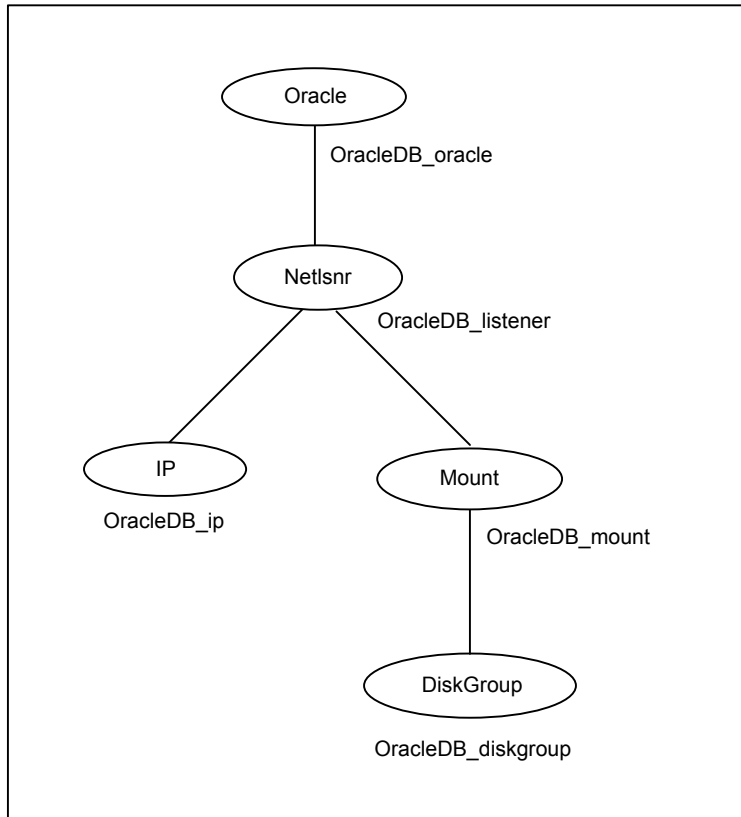
Figure A-4 shows a Service Group that manages Oracle Application Server Infrastructure components. Since the Agent for Oracle Application Server uses the OPMN daemon to start, monitor, and stop the Oracle Application Server components, you must start the OPMN component before starting any other component.

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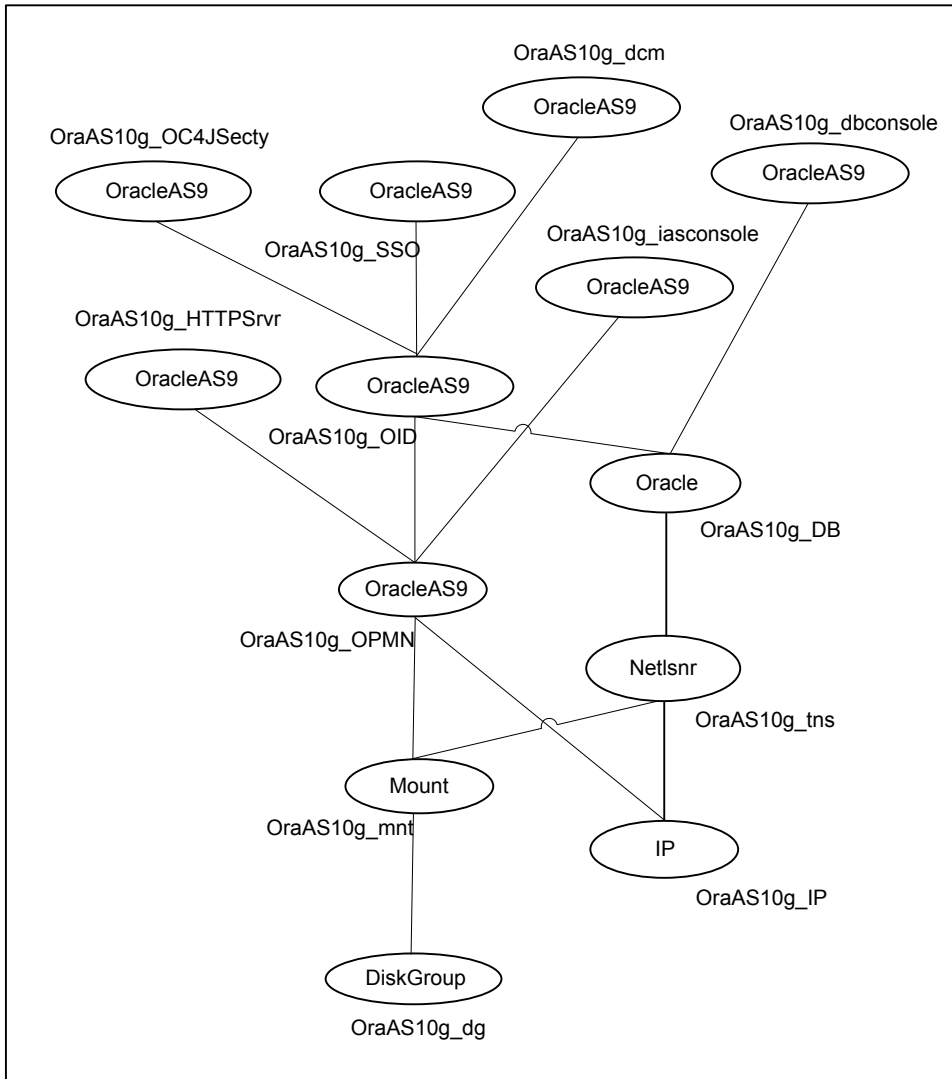
Figure A-5 and Figure A-6 show Service Groups that manage the Oracle Application Server HTTP server and Webcache services.

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
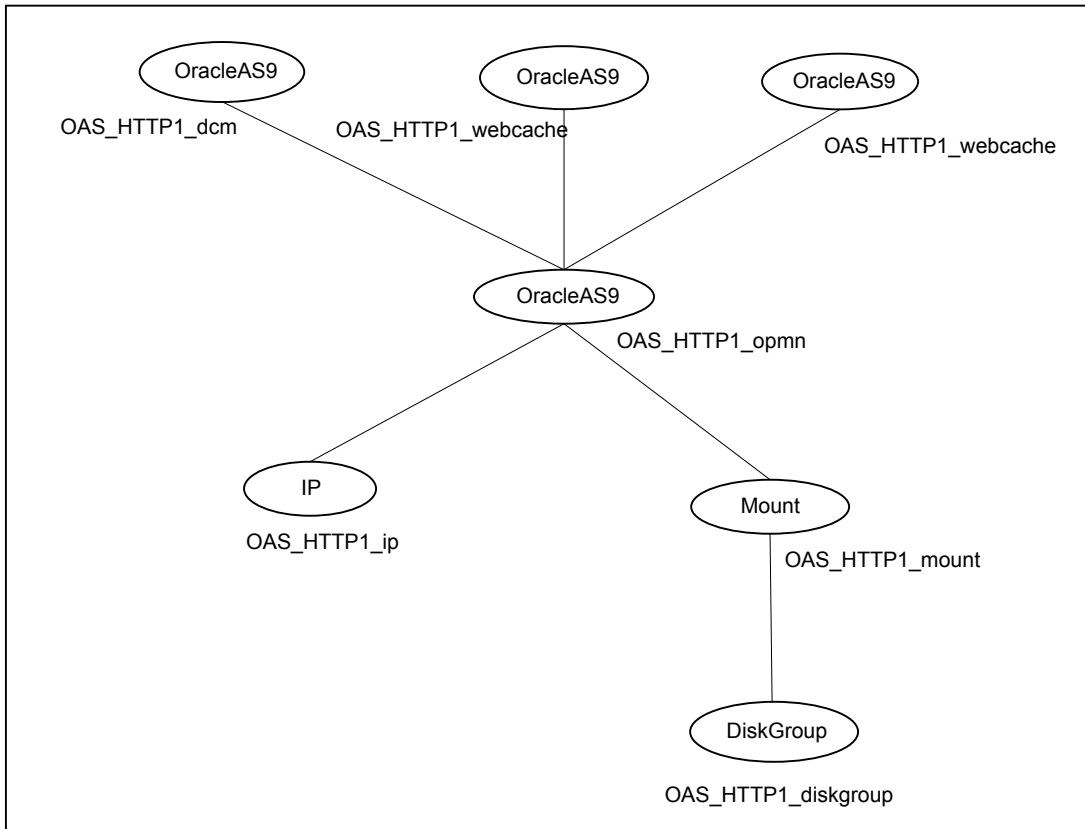
Figure A-6 Sample 2: Service Group for Oracle Application Server HTTP server and Webcache services
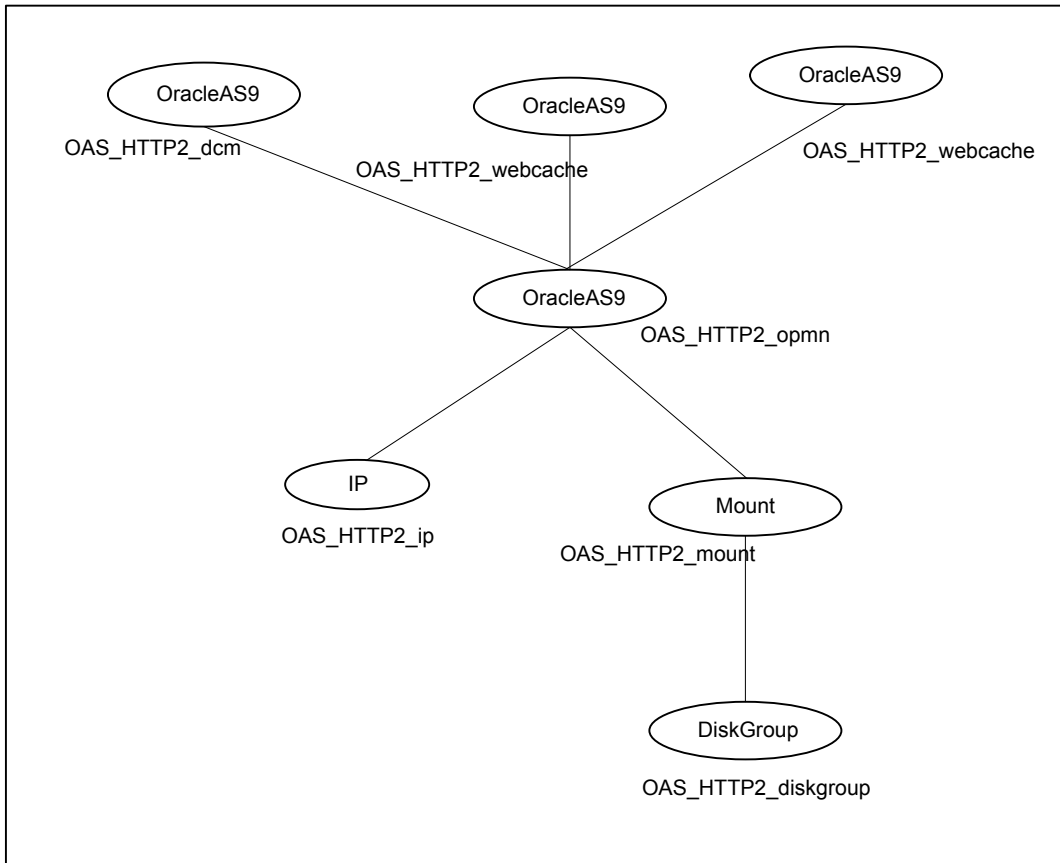
Figure A-7 and Figure A-8 show Service Groups that manage the Oracle Application Server OC4J Application Servers. Since the OPMN component is configured to start before the other Oracle Application Server components, Having multiple service groups for this function offers greater capacity as these normally 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
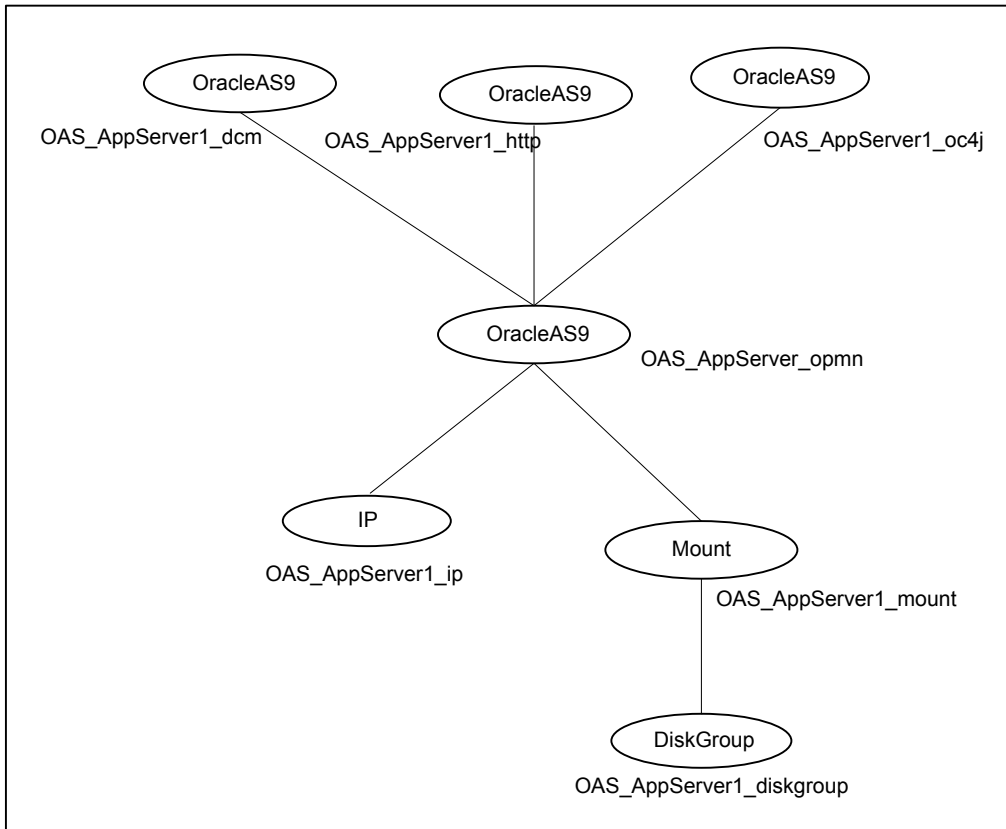
Figure A-8   Sample 2: Service Group for Oracle Application Server OC4J
              Application Server

Sample service group configuration for Oracle Fusion Middleware

The following service group configuration demonstrates how an Oracle Fusion Middleware 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-9   Oracle Fusion Middleware Service Group
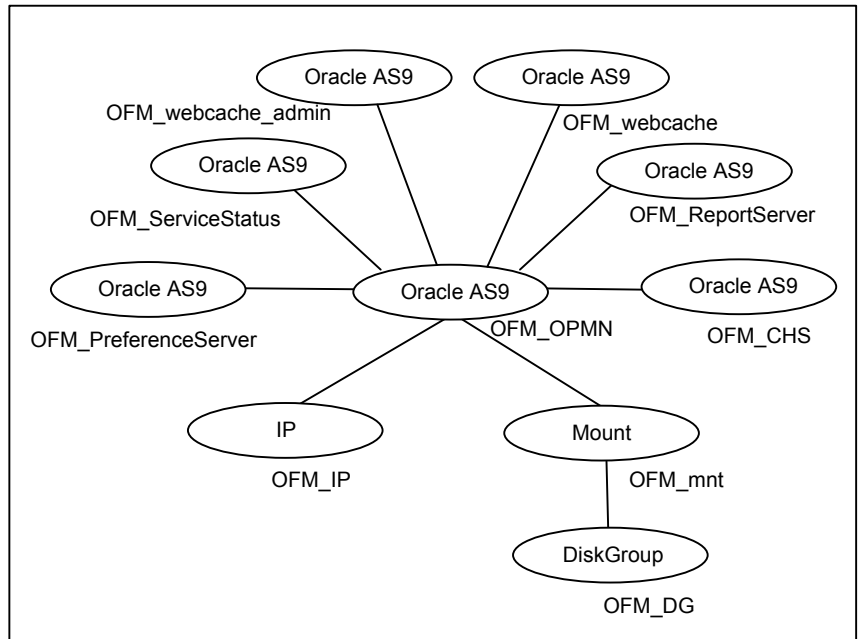
Figure A-9 shows the service group that manages the Oracle Fusion Middleware components.

The Agent for Oracle Fusion Middleware uses the OPMN daemon to start, monitor, and stop the Oracle Fusion Middleware components, you must start the OPMN component before starting any other component.

Figure A-10   EMAgent Service Group
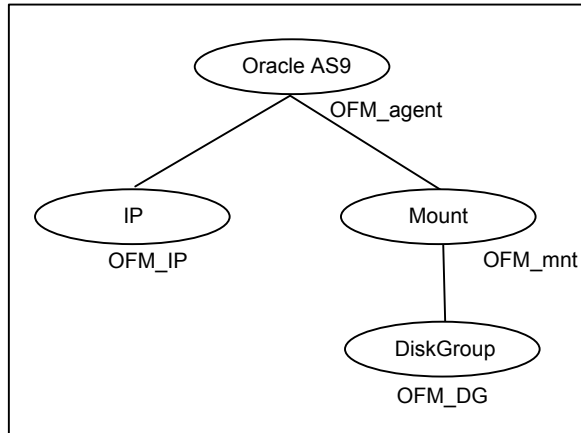
Figure A-10 shows the service group that manages the EMAgent.

Figure A-11   Oracle Identity Management Service Group
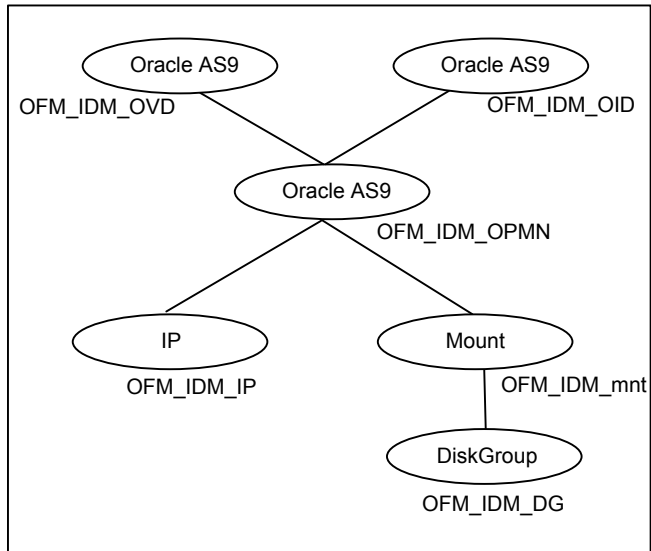
Figure A-11 shows the service group that manages the Oracle Identity Management components.

Sample Oracle Application Server resource configuration

A sample excerpt from the main.cf file is shown below for Oracle Application Server resource configuration.

```
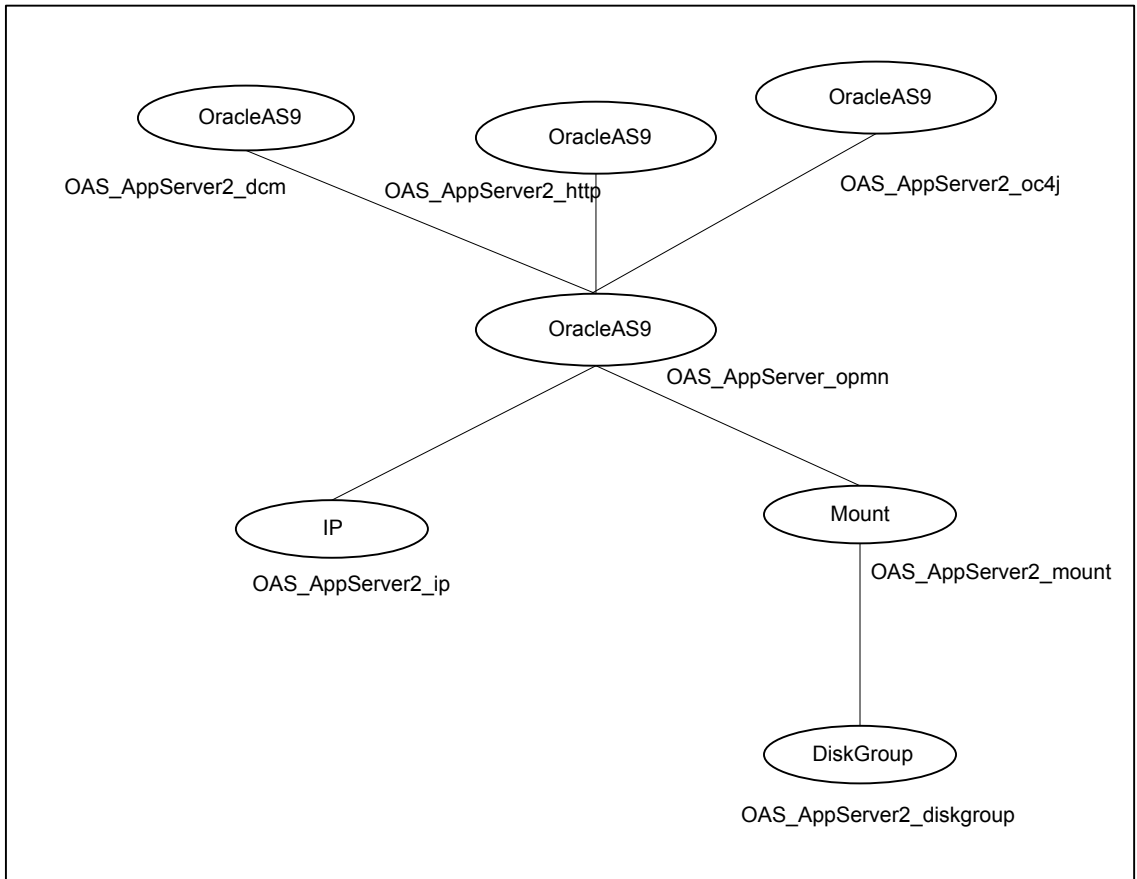group OraAS10gInfrastructure (
 SystemList = { sysA = 0, sysB = 1 }
 )

 DiskGroup OraAS10g_dg (
  DiskGroup = oradg
  )

 IP OraAS10g_IP (
  Device = bge0
  Address = "110.120.62.18"
  NetMask = "255.255.255.0"
  )

 NIC OraAS10g_NIC (
  Device = bge0
  )

 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
 )

OracleAS9 OraAS10g_ReportsServerComponent (
 ResLogLevel = TRACE
 User = oracle
 OracleHome = "/ora10g/OraHome_1"
 ComponentProcessType = ReportsServerComponent
 )
```

```
OraAS10g_IP requires OraAS10g_NIC
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
OraAS10g_ReportsServerComponent requires OraAS10g_OPMN
```

Sample Oracle Application Server resource configuration for Oracle Fusion Middleware installation

The sample Oracle Application Server resource configuration for Oracle Fusion Middleware installation is shown below.

```
include "types.cf"
include "OracleAS9Types.cf"

cluster foo (
 UserNames = { admin = gmnFmhMjnInnLvnHmk }
 Administrators = { admin }
 )

system sysA (
 )

system sysB (
 )


group ofm_grp (
 SystemList = { sysA = 0, sysB = 1 }
 )

 OracleAS9 OFM_ReportsServerComponent (
  ResLogLevel = TRACE
  User = oracle
```

```
OracleHome = "/home/oracle/OFM/OFM_HOME"
ComponentProcessType = ReportsServerComponent
OracleInstanceName = ofmnewinstance
OracleInstanceHome = "/home/oracle/OFM/OFM_New_Instance"
)

OracleAS9 OFM_agent (
ResLogLevel = TRACE
User = oracle
OracleHome = "/home/oracle/OFM/OFM_HOME"
ComponentProcessType = agent
SecondLevelMonitor = 1
OracleInstanceName = ofmnewinstance
OracleInstanceHome = "/home/oracle/OFM/OFM_New_Instance"
)

OracleAS9 OFM_OHS (
User = oracle
OracleHome = "/home/oracle/OFM/OFM_HOME"
ComponentProcessType = OHS
OracleInstanceName = ofmnewinstance
OracleInstanceHome = "/home/oracle/OFM/OFM_New_Instance"
)

OracleAS9 OFM_IDM_OID (
ResLogLevel = TRACE
User = oracle
OracleHome = "/home/oracle/OFM/IDM_HOME"
ComponentProcessType = OID
OracleInstanceName = OIDnewinstance
OracleInstanceHome = "/home/oracle/OFM/OID_New_Instance"
)

OracleAS9 OFM_OPMN (
ResLogLevel = TRACE
User = oracle
OracleHome = "/home/oracle/OFM/OFM_HOME"
ComponentProcessType = OPMN
OracleInstanceName = ofmnewinstance
OracleInstanceHome = "/home/oracle/OFM/OFM_New_Instance"
)

OracleAS9 OFM_IDM_OPMN (
```

```
     ResLogLevel = TRACE
     User = oracle
     OracleHome = "/home/oracle/OFM/IDM_HOME"
     ComponentProcessType = OPMN
     OracleInstanceName = OIDnewinstance
     OracleInstanceHome = "/home/oracle/OFM/OID_New_Instance"
     )

  OracleAS9 OFM_IDM_OVD (
     ResLogLevel = TRACE
     User = oracle
     OracleHome = "/home/oracle/OFM/IDM_HOME"
     ComponentProcessType = OVD
     OracleInstanceName = OIDnewinstance
     OracleInstanceHome = "/home/oracle/OFM/OID_New_Instance"
     )

  OracleAS9 OFM_PrefrenceServer (
     User = oracle
     OracleHome = "/home/oracle/OFM/OFM_HOME"
     ComponentProcessType = PreferenceServer
     OracleInstanceName = ofmnewinstance
     OracleInstanceHome = "/home/oracle/OFM/OFM_New_Instance"
     )

  OracleAS9 OFM_ServiceStatus (
     ResLogLevel = TRACE
     User = oracle
     OracleHome = "/home/oracle/OFM/OFM_HOME"
     ComponentProcessType = ServicesStatus
     OracleInstanceName = ofmnewinstance
     OracleInstanceHome = "/home/oracle/OFM/OFM_New_Instance"
     )

  OracleAS9 OFM_webcache (
     ResLogLevel = TRACE
     User = oracle
     OracleHome = "/home/oracle/OFM/OFM_HOME"
     ComponentProcessType = WebCache
     OracleInstanceName = ofmnewinstance
     OracleInstanceHome = "/home/oracle/OFM/OFM_New_Instance"
     )
```

```
OracleAS9 OFM_webcache_admin (
 ResLogLevel = TRACE
 User = oracle
 OracleHome = "/home/oracle/OFM/OFM_HOME"
 ComponentProcessType = WebCacheAdmin
 SecondLevelMonitor = 1
 OracleInstanceName = ofmnewinstance
 OracleInstanceHome = "/home/oracle/OFM/OFM_New_Instance"
 )

DiskGroup OFM_dg (
 DiskGroup = OFMDG
)

IP OFM_IP (
 Device = eth0
 Address = "110.120.62.18"
 NetMask = "255.255.255.0"
)

NIC OFM_NIC (
 Device = bge0
)


Mount OFM_mnt (
 MountPoint = "/OFMMNT"
 BlockDevice = "/dev/vx/dsk/OFMDG/ORAVOL"
 FSType = vxfs
 MountOpt = rw
 FsckOpt = "-y"
)

OFM_ReportsServerComponent requires OFM_OPMN
OFM_OHS requires OFM_OPMN
OFM_IDM_OID requires OFM_IDM_OPMN
OFM_IDM_OVD requires OFM_IDM_OPMN
OFM_PrefrenceServer requires OFM_OPMN
OFM_ServiceStatus requires OFM_OPMN
OFM_webcache requires OFM_OPMN
OFM_webcache_admin requires OFM_OPMN
OFM_OPMN requires OFM_mnt
OFM_OPMN requires OFM_IP
```

```
OFM_mnt requires OFM_dg
OFM_IP requires OFM_NIC
```

Sample IMF configuration

An example of a type definition file for a Oracle Application Server and Oracle Fusion Middleware agent that is IMF-enabled is as follows.

In this example, the IMF-related attributes are set to the following values:

AgentFile	/opt/VRTSvcs/bin/Script51Agent
IMF{}	{ Mode=2, MonitorFreq=3, RegisterRetryLimit=3 }
IMFRegList[]	{ User, OracleHome, ComponentProcessType }
LevelTwoMonitorFreq	25

```
type OracleAS9 (
        static str AgentDirectory = "/opt/VRTSagents/ha/bin/OracleAS9"
        static str AgentFile = "/opt/VRTSvcs/bin/Script51Agent"
        static str ArgList[] = { ResLogLevel, State, IState, User,
        OracleHome, ComponentProcessType, SecondLevelMonitor,
        MonitorProgram }
        static boolean AEPTimeout = 1
        static int LevelTwoMonitorFreq = 25
        static int IMF{} = { Mode=2, MonitorFreq=3, RegisterRetryLimit=3 }
        static str IMFRegList[] = { User, OracleHome, ComponentProcessType }
        str ResLogLevel = INFO
        str User
        str OracleHome
        str ComponentProcessType
        int SecondLevelMonitor
        str MonitorProgram
)
```

Sample service group configurations for Solaris zone support

This section includes sample service groups with Solaris zone support.

Figure A-12 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-12 Sample Service Group with Solaris zone support
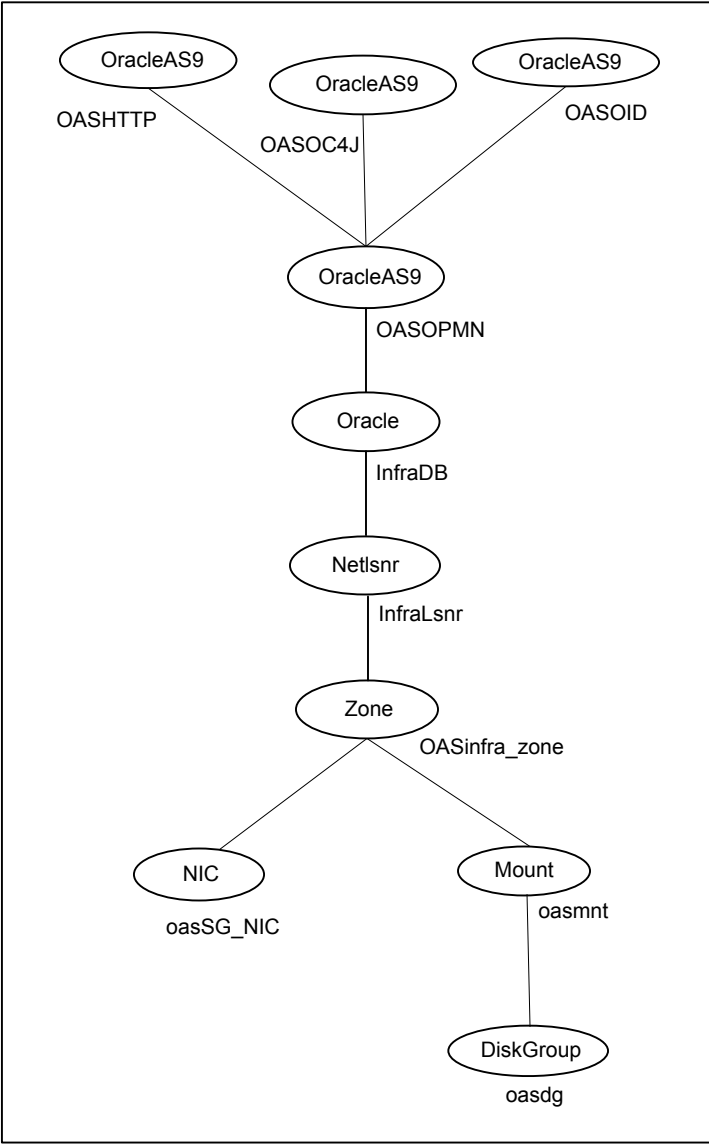
Figure A-13 shows a sample service group with Oracle HTTP Server, OC4J, and DCM running in a zone.

Figure A-13   Sample Service Group with Solaris zone support

Index