Oracle® Application Server

High Availability Guide

10*g* (10.1.4.0.1)

B28186-01

July 2006

ORACLE®

Oracle Application Server High Availability Guide, 10*g* (10.1.4.0.1)

B28186-01

Contents

3 Active-Active Topologies

4 Active-Passive Topologies

10 Deploying Identity Management with Multimaster Replication

Part IV Disaster Recovery

11 OracleAS Disaster Recovery

12 OracleAS Guard asgctl Command-line Reference

Part V Appendices

A Troubleshooting High Availability

B OracleAS Guard Error Messages

Preface

This preface contains these sections:

Audience

The *Oracle Application Server High Availability Guide* is intended for administrators, developers, and others whose role is to deploy and manage Oracle Application Server with high availability requirements.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at

`http://www.oracle.com/accessibility/`

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

Related Documentation

For more information, see these Oracle resources:

- *Oracle Application Server Installation Guide*
- *Oracle Application Server Administrator's Guide*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Part I

Overview

The chapters in this part provide an introduction to Oracle Application Server high availability.

This part contains the following chapter:

- Chapter 1, "Introduction to High Availability"

1 Introduction to High Availability

This release of Oracle Application Server extends and improves upon the high availability solutions that were available in earlier releases. New, flexible, and automated high availability solutions for Oracle Application Server have been tested and are described in this guide. All of these solutions seek to ensure that applications that you deploy on Oracle Application Server meet the required availability to achieve your business goals. The solutions and procedures described in this book seek to eliminate single points of failure of any Oracle Application Server components with no or minimal outage in service.

This chapter explains high availability and its importance from the perspective of Oracle Application Server.

This chapter contains the following sections:

1.1 What Is High Availability

This section provides an overview of high availability from a problem-solution perspective. It contains these sections:

1.1.1 High Availability Problems

Mission critical computer systems need to be available 24 hours a day, 7 days a week, and 365 days a year. However, part or all of the system may be down during planned or unplanned downtime. A system's availability is measured by the percentage of time that it is providing service in the total time since it is deployed. Table 1–1 provides an example.

Table 1–1 Availability Percentages and Corresponding Downtime Values

Availability Percentage	Approximate Downtime Per Year
95%	18 days
99%	4 days

Table 1–1 (Cont.) Availability Percentages and Corresponding Downtime Values

Availability Percentage	Approximate Downtime Per Year
99.9%	9 hours
99.99%	1 hour
99.999%	5 minutes

Table 1–2 depicts the various types of failures that are possible with a computer system.

Table 1–2 System Downtime and Failure Types

Downtime Type	Failure Type
Unplanned downtime	System failure
	Data failure
	Disasters
	Human error
Planned downtime	System maintenance[1]
	Data maintenance

[1] Includes hardware and/or software changes (operating system, application server, configuration, application changes).

These two types of downtimes (planned and unplanned) are usually considered separately when designing a system's availability requirements. A system's needs may be very restrictive regarding its unplanned downtimes, but very flexible for planned downtimes. This is the typical case for applications with high peak loads during working hours, but that remain practically inactive at night and during weekends.

1.1.2 High Availability Solutions

Oracle Application Server provides both local high availability and disaster recovery solutions for maximum protection against any kind of failure with flexible installation, deployment, and security options. The redundancy of Oracle Application Server local high availability and disaster recovery originates from its redundant high availability architectures.

High availability solutions can be categorized into:

- local high availability solutions, which provide high availability in a single data center deployment. Local high availability solutions protect against failures such as process, node, and media failures as well as human errors.

  Oracle Application Server provides many local high availability solutions. See Chapter 2, "Overview of Oracle Application Server High Availability Topologies" for an overview.

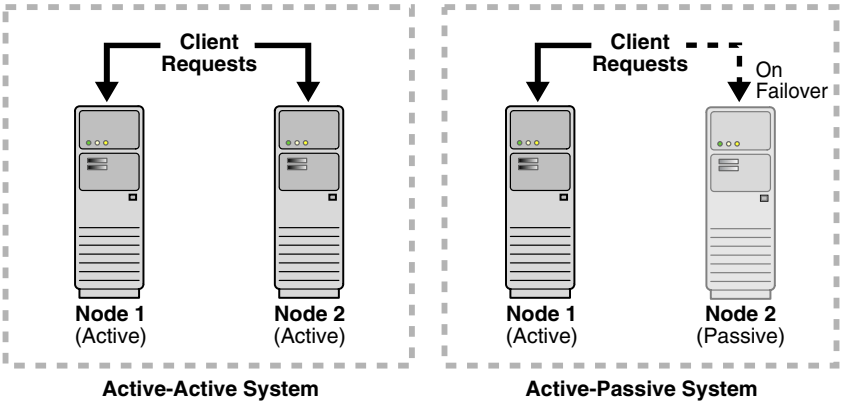
- disaster recovery solutions, which are usually geographically distributed deployments that protect your applications from disasters such as floods or regional network outages.

1.1.2.1 Local High Availability Solutions

To solve the high availability problem, a number of technologies and best practices are needed. The most important mechanism is redundancy. High availability comes from

redundant systems and components. Local high availability solutions can be categorized, by how they provide redundancy, into active-active solutions and active-passive solutions (see Figure 1–1).

Figure 1–1 Active-Active and Active-Passive High Availability Solutions



Active-active solutions deploy two or more active system instances and can be used to improve scalability as well as provide high availability. All instances handle requests concurrently.

Active-passive solutions deploy an active instance which handles requests and a passive instance which is on standby. A heartbeat mechanism is set up between these two instances. This mechanism is provided and managed through operating system vendor-specific clusterware. Generally, vendor-specific cluster agents are also available to automatically monitor and failover between cluster nodes, so that when the active instance fails, an agent shuts down the active instance completely, brings up the passive instance, and application services can successfully resume processing. As a result, the active-passive roles are now switched. The same procedure can be done manually for planned or unplanned downtime. Active-passive solutions are also generally referred to as cold failover clusters.

From the entry point of an Oracle Application Server system (content cache) to the back end layer (data sources), all the tiers that are crossed by a client request can be configured in a redundant manner either in an active-active topology using OracleAS Clusters or in an active-passive topology using OracleAS Cold Failover Clusters.

In addition to architectural redundancies, the following local high availability technologies are also necessary in a comprehensive high availability system:

- Process death detection and automatic restart

  Processes may die unexpectedly due to configuration or software problems. A proper process monitoring and restart system should monitor all system processes constantly and restart them should problems appear.

  A system process should also maintain the number of restarts within a specified time interval. This is also important since continually restarting within short time periods may lead to additional faults or failures. Therefore a maximum number of restarts or retries within a specified time interval should also be designed as well.

  Oracle Application Server performs process death detection and automatic restart through Oracle Process Manager and Notification Server (OPMN). For details, see Section 1.3.1, "Process Death Detection and Automatic Restart".

- Clustering

Clustering components of a system together enables the components to be viewed functionally as a single entity from the perspective of a client for runtime processing and manageability. A cluster is a set of processes running on single or multiple nodes that share the same workload. There is a close correlation between clustering and redundancy. A cluster provides redundancy for a system.

Oracle Application Server provides clustering at different levels (for example, instance clustering and component clustering). For creating high availability topologies, you can cluster at the instance level with OracleAS Clusters. See Chapter 2, "Overview of Oracle Application Server High Availability Topologies" for details.

- Configuration management

  A cluster of components often need to share common configuration. Proper configuration management ensures that components provide the same reply to the same incoming request, enables the components to synchronize their configurations, and provides highly available configuration management for less administration downtime.

  In Oracle Application Server, the Distributed Configuration Management (DCM) component synchronizes configuration information for the members of a cluster.

- State replication and routing

  For stateful applications, client state can be replicated to enable stateful failover of requests in the event that processes servicing these requests fail.

- Server load balancing and failover

  When multiple instances of identical server components are available, client requests to these components can be load balanced to ensure that the instances have roughly the same workload. With a load balancing mechanism in place, the instances are redundant. If any of the instances fail, requests to the failed instance can be sent to the surviving instances.

  Oracle Application Server components perform load balancing as necessary when communicating with other components. See Section 1.3.2, "Internal Load Balancing in Oracle Application Server" for details.

- Backup and recovery

  User errors may cause a system to malfunction. In certain circumstances, a component or system failure may not be repairable. A backup and recovery facility should be available to back up the system at certain intervals and restore a backup when an unrepairable failure occurs.

  Oracle Application Server provides the OracleAS Backup and Recovery Tool to assist you in backing up and restoring Oracle Application Server files. See Section 1.3.3, "Backup and Recovery" for details.

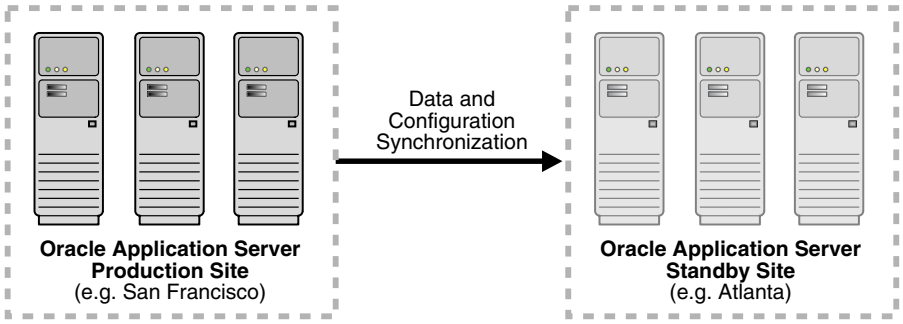
1.1.2.2 Disaster Recovery Solutions

The Oracle Application Server Disaster Recovery solution builds on top of the local high availability solutions and includes Oracle Application Server Guard. This unique solution combines the proven Oracle Data Guard technology in the Oracle Database with advanced disaster recovery technologies in the application realm to create a comprehensive disaster recovery solution. This solution requires homogenous production and standby sites, but other Oracle Application Server instances can also be installed in either site as long as they do not interfere with the instances in the disaster recovery setup. Configurations and data must be synchronized regularly between the two sites to maintain homogeneity.

Disaster recovery solutions typically set up two homogeneous sites, one active and one passive. Each site is a self-contained system. The active site is generally called the production site, and the passive site is called the standby site. During normal operation, the production site services requests; in the event of a site failover or switchover, the standby site takes over the production role and all requests are routed to that site. To maintain the standby site for failover, not only must the standby site contain homogeneous installations and applications, data and configurations must also be synchronized constantly from the production site to the standby site.

See Part IV, "Disaster Recovery" for complete details on OracleAS Disaster Recovery solutions.

Figure 1–2 Geographically Distributed Disaster Recovery



1.2 Oracle Application Server High Availability Concepts

The following sections provide an overview of high availability in Oracle Application Server:

- Section 1.2.1, "Terminology"
- Section 1.2.2, "Oracle Application Server Base Architecture"
- Section 1.2.3, "Oracle Application Server Base Architecture with Oracle Access Manager"
- Section 1.2.4, "Oracle Application Server Base Architecture with Oracle Identity Federation"

1.2.1 Terminology

This book uses the terms below to describe high availability concepts and topologies:

- **active-active**: In an active-active high availability system, the equivalent members in that system can service requests concurrently. Under normal operation where none of the members have failed, all members are active and none are on standby.
- **active-passive**: In an active-passive high availability system, some members of the system can be actively servicing requests and performing work, while other members are inactive (passive). They become active only if one or more of the active nodes have failed. Consumers of services provided by the system may or may not notice the failure. An active-active system generally provides more transparency and options for scalability to consumers than an active-passive system.
- **failover**: When a member of a highly available system fails unexpectedly (unplanned downtime), in order to continue offering services to its consumers, the system undergoes a failover operation. If the system is an active-passive system,

the passive member is activated during the failover operation and consumers are directed to it instead of the failed member. The failover process can be performed manually, or it can be automated by setting up hardware cluster services to detect failures and move cluster resources from the failed node to the standby node. If the system is an active-active system, the failover is performed by the load balancer entity serving requests to the active members. If an active member fails, the load balancer detects the failure and automatically redirects requests for the failed member to the surviving active members.

- **failback**: After a system undergoes a successful failover operation, the original failed member can be repaired over time and be re-introduced into the system as a standby member. If desired, a failback process can be initiated to activate this member and deactivate the other. This process reverts the system back to its pre-failure configuration.
- **hardware cluster**: A hardware cluster is a collection of nodes that provides a single view of network services (for example: an IP address) or application services (for example: databases, Web servers) to clients of these services. Each node in a hardware cluster is a standalone server that runs its own processes. These processes can communicate with one another to form what looks like a single system that cooperatively provides applications, system resources, and data to users.

  A hardware cluster achieves high availability and scalability through the use of specialized hardware (cluster interconnect, shared storage) and software (health monitors, resource monitors). (The cluster interconnect is a private link used by the hardware cluster for heartbeat information to detect node death.) Due to the need for specialized hardware and software, hardware clusters are commonly provided by hardware vendors such as Sun, Hewlett-Packard, IBM, and Dell. While the number of nodes that can be configured in a hardware cluster is vendor dependent, for the purpose of Oracle Application Server high availability, only two nodes are required. Hence, this document assumes a two-node hardware cluster for high availability solutions employing a hardware cluster.
- **cluster agent**: A cluster agent is the software that runs on a node member of a hardware cluster that coordinates availability and performance operations with other nodes. Clusterware provides resource grouping, monitoring, and the ability to move services. A cluster agent can automate the service failover.
- **clusterware**: Clusterware is the software that manages the operations of the members of a cluster as a system. It enables you to define resources and services to monitor via a heartbeat mechanism between cluster members and to move these resources and services to a different member in the cluster as efficiently and transparently as possible.
- **shared storage**: Although each hardware cluster node is a standalone server that runs its own set of processes, the storage subsystem required for any cluster-aware purpose is usually shared. Shared storage refers to the ability of the cluster to be able to access the same storage, usually disks, from both the nodes. While the nodes have equal access to the storage, only one node, the primary node, has active access to the storage at any given time. The hardware cluster's software grants the secondary node access to this storage if the primary node fails. For the OracleAS Infrastructure in the OracleAS Cold Failover Cluster environment, its `ORACLE_HOME` is on such a shared storage file system. This file system is mounted by the active node; if that node fails, the standby node takes over and mounts the file system. In some cases, the active node may relinquish control of the shared storage, such as when the hardware cluster's software deems the OracleAS

Infrastructure as unusable from the active node and decides to move it to the standby node.

- **primary node**: The primary node is the node that is actively running one or more Oracle Application Server installations at any given time. If this node fails, Oracle Application Server is failed over to the secondary node. Because the primary node runs the active Oracle Application Server installation(s), it is considered the "hot" node. See the definition for "secondary node" in this section.
- **secondary node**: The secondary node is the node that takes over the execution of Oracle Application Server if the primary node fails. Because the secondary node does not originally run Oracle Application Server, it is considered the "cold" node. Because Oracle Application Server fails over from a hot node (primary) to a cold node (secondary), this type of failover is called cold failover. See the definition for "primary node" in this section.
- **network hostname**: The network hostname is a name assigned to an IP address through the `/etc/hosts` file (in UNIX), `C:\WINDOWS\system32\drivers\etc\hosts` file (in Windows), or through DNS resolution. This name is visible in the network to which the machine is connected. The network hostname and the physical hostname are usually the same. However, each machine has only one physical hostname but may have multiple network hostnames. Thus, a machine's network hostname may not always be its physical hostname.
- **physical hostname**: This guide differentiates between physical hostname and network hostname. This guide uses physical hostname to refer to the "internal name" of the current machine. In UNIX, this is the name returned by the `hostname` command.

  Physical hostname is used by Oracle Application Server middle-tier installation types to reference the local host. During installation, the installer automatically retrieves the physical hostname from the current machine and stores it in the Oracle Application Server configuration metadata on disk.
- **switchover**: During normal operation, active members of a system may require maintenance or upgrading. A switchover process can be initiated to enable a substitute member to take over the workload performed by the member that requires maintenance or upgrading, which undergoes planned downtime. The switchover operation ensures continued service to consumers of the system.
- **switchback**: When a switchover operation is performed, a member of the system is deactivated for maintenance or upgrading. When the maintenance or upgrading is completed, the system can undergo a switchback operation to activate the upgraded member and bring the system back to the pre-switchover configuration.
- **virtual hostname**: Virtual hostname is a network addressable hostname that maps to one or more physical machines via a load balancer or a hardware cluster. For load balancers, the "virtual server name" is used interchangeably with virtual hostname in this book. A load balancer can hold a virtual hostname on behalf of a set of servers, and clients communicate indirectly with the machines using the virtual hostname. A virtual hostname in a hardware cluster is a network hostname assigned to a cluster virtual IP. Because the cluster virtual IP is not permanently attached to any particular node of a cluster, the virtual hostname is not permanently attached to any particular node either.

> **Note:** Whenever the phrase "virtual hostname" is used in this guide, it is assumed to be associated with a virtual IP address. In cases where just the IP address is needed or used, it will be explicitly stated.

- **virtual IP, cluster virtual IP**: Generally, a virtual IP can be assigned to a hardware cluster or load balancer. To present a single system view of a cluster to network clients, a virtual IP serves as an entry point IP address to the group of servers which are members of the cluster. A virtual IP can be assigned to a server load balancer or a hardware cluster.

  A hardware cluster uses a cluster virtual IP to present to the outside world the entry point into the cluster (it can also be set up on a standalone machine). The hardware cluster's software manages the movement of this IP address between the two physical nodes of the cluster while clients connect to this IP address without knowing which physical node this IP address is currently active on. In a typical two-node hardware cluster topology, each machine has its own physical IP address and physical hostname, while there could be several cluster IP addresses. These cluster IP addresses float or migrate between the two nodes. The node with current ownership of a cluster IP address is active for that address.

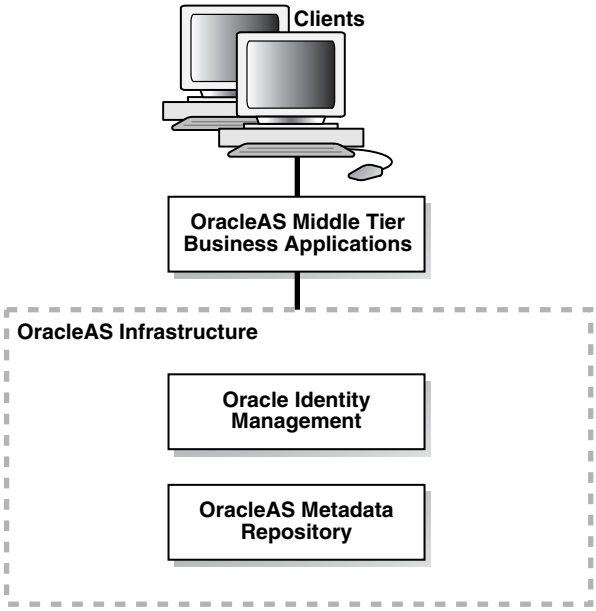
  A load balancer also uses a virtual IP as the entry point to a set of servers. These servers tend to be active at the same time. This virtual IP address is not assigned to any individual server but to the load balancer which acts as a proxy between servers and their clients.

1.2.2 Oracle Application Server Base Architecture

Before building Oracle Application Server high availability topologies, you should understand Oracle Application Server's base architecture because to create a highly available topology, you add redundancy to the base architecture. This means that you add redundancy to all Oracle Application Server components and also to connections between components.

Figure 1–3 illustrates the base architecture of Oracle Application Server.

Figure 1–3 Oracle Application Server Base Architecture



At a high level, Oracle Application Server consists of the Oracle Application Server middle-tier business applications, Oracle Identity Management, and OracleAS Metadata Repository. The latter two are part of the OracleAS Infrastructure.

Oracle Identity Management manages user authentication, authorization, and identity information. It includes the following components:

- OracleAS Single Sign-On
- Oracle Delegated Administration Services
- Oracle Internet Directory
- Oracle Directory Integration Platform

Architecturally, Oracle Identity Management can be broken down into a Web server tier of Oracle HTTP Server, an OracleAS Single Sign-On/Oracle Delegated Administration Services middle-tier composed of an Oracle Containers for J2EE (OC4J) instance for these security applications, and an Oracle Internet Directory/Oracle Directory Integration Platform tier at the back end. OracleAS Metadata Repository is an Oracle database that manages configuration, management, and product metadata for middle-tier and Oracle Identity Management components.

The middle tier hosts most of Oracle Application Server business applications, such as:

- Oracle Application Server Portal
- Oracle Application Server Wireless
- Oracle Application Server Integration

These applications rely on Oracle Identity Management and OracleAS Metadata Repository for security and metadata support. The middle tier also includes a Web caching sub-tier (Oracle Application Server Web Cache), a Web server sub-tier (Oracle HTTP Server), and OC4J instances. Behind the middle tier, the OracleAS Metadata Repository serves as the data tier. In actual deployments, other databases may also exist in the data tier (for example, a customer database for OC4J applications deployed on the middle tier).

Figure 1–4 shows the various tiers that are traversed by client requests to the Oracle Application Server business applications and the Oracle Application Server Infrastructure services. Figure 1–5 provides an overall view of Oracle Identity Management, OracleAS Metadata Repository, and LDAP services.

Although Oracle Application Server provides many features that make it easy to build high availability topologies (such as automatic process monitoring and restart, and backup and recovery), they do not provide complete high availability. Several single points of failure exist. To eliminate them, you need to provide redundancy for each component. This can be achieved by extending the base architecture with additional high availability architectures.

Figure 1–4 Base Architecture of Oracle Application Server

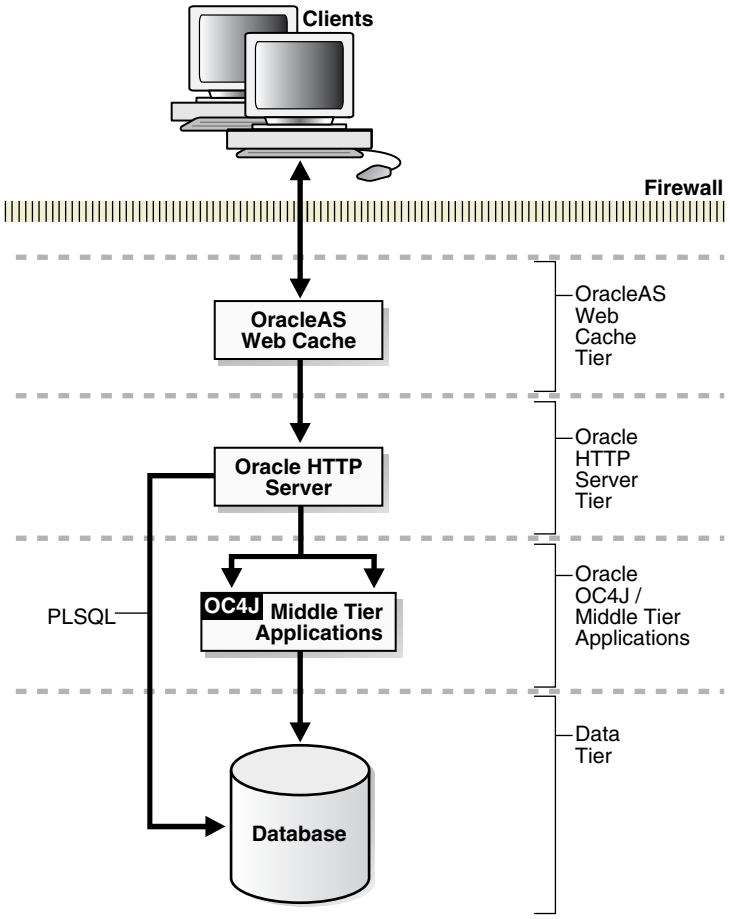
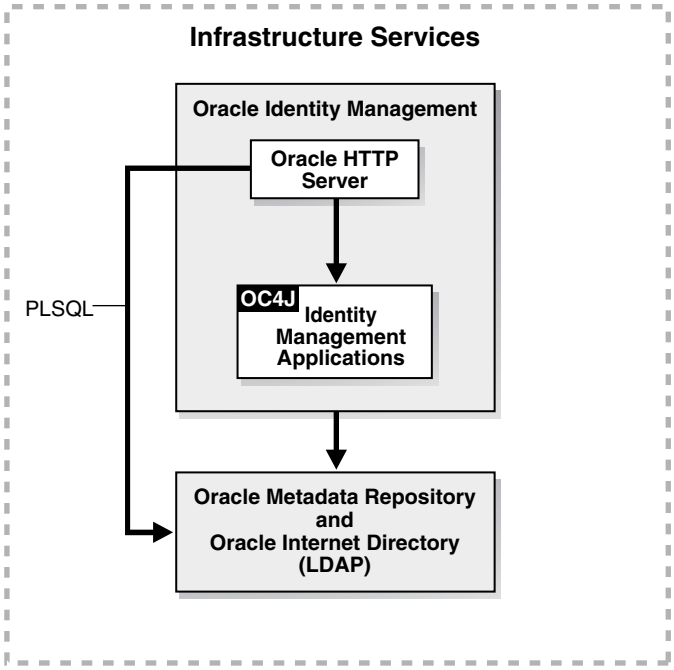


Figure 1–5 Overview of Infrastructure Services



1.2.3 Oracle Application Server Base Architecture with Oracle Access Manager

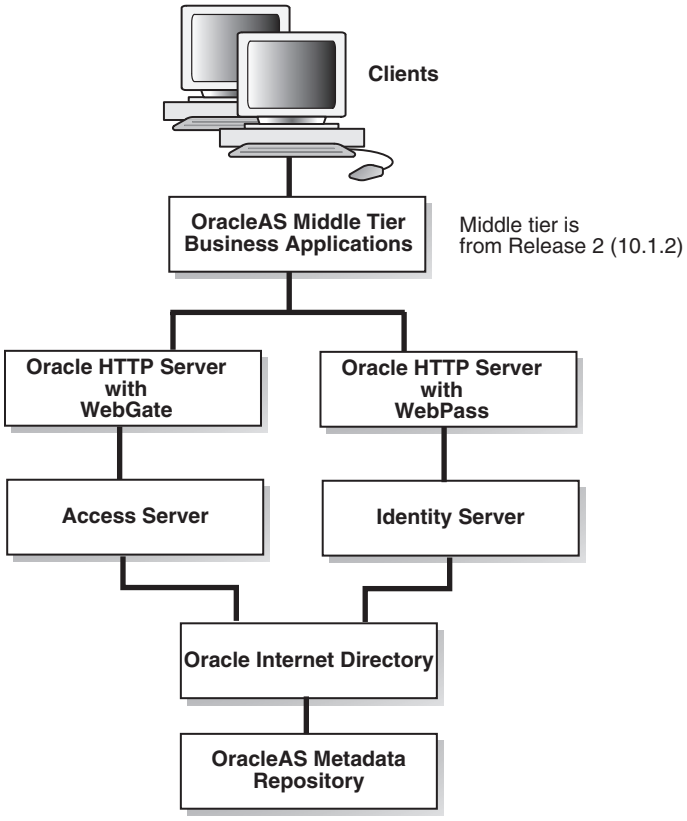
Oracle Access Manager provides single sign-on, authorization, and authentication services for your applications. It is an alternative to using OracleAS Single Sign-On and Oracle Delegated Administration Services.

Oracle Access Manager stores its information on users in an LDAP directory server (for example, Oracle Internet Directory).

Oracle Access Manager includes WebGate and WebPass, which are plug-ins for the Oracle HTTP Server. WebGate directs requests from the Oracle HTTP Server to the Access Server, and WebPass directs requests from the Oracle HTTP Server to the Identity Server.

Figure 1–6 shows a simple topology that involves Oracle Access Manager.

Figure 1–6 Simple Topology with Oracle Access Manager



1.2.4 Oracle Application Server Base Architecture with Oracle Identity Federation

Oracle Identity Federation goes beyond simple single sign-on in that it enables you to share identity information with your business partners. In a federated environment (that is, an environment where services and information are shared between trusted partners), users have to sign on only once to access all the entities in the federation. Without Oracle Identity Federation, users have to sign on at each entity.

1.3 Oracle Application Server Features Related to High Availability

Oracle Application Server includes the following features that are helpful in high availability topologies. These features are used in all Oracle Application Server topologies, not just high availability topologies.

- Section 1.3.1, "Process Death Detection and Automatic Restart"
- Section 1.3.2, "Internal Load Balancing in Oracle Application Server"
- Section 1.3.3, "Backup and Recovery"

1.3.1 Process Death Detection and Automatic Restart

An Oracle Application Server instance runs many different processes to serve client requests. Ensuring high availability means ensuring that all these processes run smoothly, fulfill requests, and do not experience any unexpected hangs or failures.

The interdependency of these processes must also be managed so that they are started in the proper sequence, with processes starting up only after the processes that they are dependent on have started successfully.

Oracle Application Server provides management services at the process level with Oracle Process Manager and Notification Server (OPMN). OPMN has the following capabilities:

- Provides automatic death detection of Oracle Application Server processes.
- Provides automatic restart of Oracle Application Server processes when they become unresponsive, terminate unexpectedly, or become unreachable as determined by ping and notification operations.
- Channels events from different Oracle Application Server component instances to all Oracle Application Server components that can utilize them.
- Enables gathering of host and Oracle Application Server process statistics and tasks.
- Does not depend on any other Oracle Application Server component being up and running before it can be started and used.

OPMN manages the following Oracle Application Server components:

- Oracle HTTP Server
- Oracle Containers for J2EE (OC4J)
- Distributed Configuration Management daemon
- OracleAS Log Loader
- OracleAS Guard (for disaster recovery)
- Oracle Internet Directory
- OracleAS Port Tunnelling

In addition, OPMN implicitly manages any applications that rely on the above components. For example, J2EE applications, which run under OC4J, are managed by OPMN.

> **Note:** OPMN does not manage Oracle Access Manager or Oracle Identity Federation.

OPMN running on different Oracle Application Server instances can also work together to provide distributed process management and control. For example, you can issue an OPMN command on one machine to start all processes or a specific process type across all local and remote Oracle Application Server instances.

OPMN consists of two major components:

- Oracle Notification Server (ONS)

  The ONS is the transport mechanism for failure, recovery, startup, and other related notifications between components in Oracle Application Server. It operates according to a publish-subscribe model: an Oracle Application Server component receives a notification of a certain type per its subscription to ONS. When such a notification is published, ONS sends it to the appropriate subscribers.

- Oracle Process Manager (PM)

The PM is the centralized process management mechanism in Oracle Application Server and is used to manage Oracle Application Server processes. It is responsible for starting, restarting, stopping, and monitoring every process it manages. The PM handles all requests sent to OPMN associated with controlling a process or obtaining status about a process. The PM is also responsible for performing death-detection and automatic restart of the processes it manages. The Oracle Application Server processes that the PM is configured to manage are specified in a file named `opmn.xml`. The PM waits for a user command to start processes. When a specific process or all processes are to be stopped, the PM receives a request as specified by the request parameters.

1.3.2 Internal Load Balancing in Oracle Application Server

Load balancing involves the ability to distribute requests among two or more processes. A software or hardware load balancer typically includes the following features:

- load balancing algorithm

  A set of rules for distributing requests across the different instances. Common load balancing algorithms include simple round-robin or assignment based on some weighted property of the instance, such as the response time or capacity of that instance relative to other instances.

- death detection

  The ability to recognize failed requests to one or more instances, and additionally, the ability to mark those instances as inactive so that no further requests will be forwarded to them.

Oracle Application Server uses different load balancing mechanisms to communicate requests between components in an Oracle Application Server system. Load balancing takes place:

- from OracleAS Web Cache to Oracle HTTP Servers
- from Oracle HTTP Servers to OC4J processes for J2EE applications
- from Oracle HTTP Servers to the database for PL/SQL applications
- intra OC4J processes from the presentation layer components (servlets and JSPs) to the business layer components (EJBs)
- from OC4J processes to databases
- from WebGate to Access Servers
- from WebPass to Identity Servers

All tiers in Oracle Application Server are enabled to manage failures in the connections that they establish with the next tier as follows:

- Connections established from OracleAS Web Cache to Oracle HTTP Servers: OracleAS Web Cache detects failures in the replies returned by Oracle HTTP Servers and routes the new requests to the available Oracle HTTP Servers.
- Connections established from Oracle HTTP Servers to OC4J processes: Oracle HTTP Server maintains a routing table of available OC4J processes and routes new requests only to those OC4J processes that are up an running.
- Connections established from Oracle HTTP Servers to databases: mod_plsql detects failures in the database and routes requests to the available database nodes.

- Connections established between OC4J processes: OC4J detects failures in the RMI invocations to the EJB tier and fails communication over to available EJB nodes.
- Connections established between OC4J processes and databases: OC4J drivers are enabled to detect failures of database nodes and re-route requests to available nodes.

1.3.3 Backup and Recovery

Backing up your files to protect against data loss is critical to maintaining a highly available environment. Oracle Application Server includes the OracleAS Backup and Recovery Tool to help you back up configuration and data files.

You can use the OracleAS Backup and Recovery Tool to back up and recover the following types of files:

- configuration files in the middle-tier and OracleAS Infrastructure homes
- OracleAS Metadata Repository files

The OracleAS Backup and Recovery Tool is installed whenever you install Oracle Application Server. The tool is installed in the `ORACLE_HOME/backup_restore` directory.

The OracleAS Backup and Recovery Tool supports the following installation types:

- OracleAS Infrastructure (Identity Management and Metadata Repository)
- OracleAS Infrastructure (Identity Management only)
- OracleAS Infrastructure (Metadata Repository only)

> **Note:** OracleAS Backup and Recovery Tool does not back up files in the Oracle Access Manager or Oracle Identity Federation homes.

A complete Oracle Application Server environment backup includes:

- A full backup of all files in the middle-tier Oracle homes (this includes Oracle software files and configuration files).
- A full backup of all files in the OracleAS Infrastructure home (this includes Oracle software files and configuration files).
- A complete cold backup of the OracleAS Metadata Repository.
- A full backup of the Oracle system files on each host in your environment.

See the "Backup and Recovery" chapters in the *Oracle Application Server Administrator's Guide* for details.

1.4 High Availability Information in Other Documentation

Table 1–3 provides a list of cross-references to high availability information in other documents in the Oracle library. This information mostly pertains to high availability of various Oracle Application Server components.

Table 1–3 High Availability Information in Oracle Documentation

Component	Location of Information
Oracle installer	In the chapters for installing in high availability environments in *Oracle Application Server Installation Guide*.
Oracle Application Server Backup and Recovery Tool	In the backup and restore part of *Oracle Application Server Administrator's Guide*.
Identity Management service replication	In the "Advanced Configurations" chapter of *Oracle Application Server Single Sign-On Administrator's Guide*.
Identity Management high availability deployment	In the "Identity Management Deployment Planning" chapter of *Oracle Identity Management Infrastructure Administrator's Guide*.
Database high availability	*Oracle High Availability Architecture and Best Practices*
Oracle Process Manager and Notification Server commands	*Oracle Process Manager and Notification Server Administrator's Guide*
Load balancing to OC4J processes	*Oracle HTTP Server Administrator's Guide*

In addition, references to these and other documentation are noted in the text of this guide, where applicable.

Part II

High Availability Topologies

The chapters in this part describe high availability topologies in Oracle Application Server.

This part contains the following chapters:

2

Overview of Oracle Application Server High Availability Topologies

Oracle Application Server local high availability topologies include several active-active and active-passive topologies. Within each type of topology, multiple solutions exist that differ in ease of installation, cost, scalability, and security.

This chapter contains the following sections:

2.1 About the High Availability Topologies

You can group Oracle Application Server high availability topologies into two categories: active-active and active-passive: Table 2–1 summarizes the topology types.

Table 2–1 High Availability Topology Types

Topology	Description
Active-Active	In active-active topologies, you run multiple active Oracle Application Server instances on multiple nodes. All of the instances are servicing requests concurrently. If an instance or a node fails, the remaining active instances take over the workload of the failed instance. Active-active topologies use an external load balancer to distribute requests to the active instances. Active-active topologies for Oracle Application Server are also called OracleAS Clusters topologies. Specifically, active-active topologies for Oracle Identity Management are also called "OracleAS Cluster (Identity Management)" topologies. Active-active topologies are: ■ OracleAS Cluster (Identity Management) Topology ■ Distributed OracleAS Cluster (Identity Management) Topology

Table 2–1 (Cont.) High Availability Topology Types

Topology	Description
Active-Passive	In active-passive topologies, you have two nodes in a hardware cluster, but only one of the nodes is active at any time. If the active node or instance fails, the passive node becomes active and takes over the entire workload of the failed instance. The active and passive nodes share a storage device, on which you install Oracle Application Server. The nodes also use a virtual hostname, through which you access the active node. If the active node fails, the virtual hostname points to the other node, which becomes the active node. Active-passive topologies in Oracle Application Server are also called OracleAS Cold Failover Cluster topologies. Active-passive topologies are: ■ OracleAS Cold Failover Cluster (Infrastructure) Topology ■ Distributed OracleAS Cold Failover Cluster (Infrastructure) Topology ■ OracleAS Cold Failover Cluster (Identity Management) Topology ■ Distributed OracleAS Cold Failover Cluster (Identity Management) Topology

Although all topologies provide high availability, active-active solutions generally offer higher scalability and faster failover. On the downside, they tend to be more expensive.

For all topologies, you can install and run the Oracle Application Server components together (that is, from the same Oracle home), or you can install and run them in a distributed manner (that is, on separate nodes and separate Oracle homes).

Table 2–2 compares how components are distributed in the Oracle Application Server topologies. Later chapters in this guide provide details.

Table 2–2 Distribution of Components in High Availability Topologies

Topology	OracleAS Metadata Repository	Oracle Identity Management
OracleAS Cluster (Identity Management) Topology	Installed in an existing database.	Installed in an active-active configuration.
Distributed OracleAS Cluster (Identity Management) Topology	Installed in an existing database.	**Oracle Internet Directory** and **Oracle Directory Integration Platform**: installed in an active-active configuration. **OracleAS Single Sign-On** and **Oracle Delegated Administration Services**: installed in an active-active configuration.
OracleAS Cold Failover Cluster (Infrastructure) Topology	Installer installs the Oracle Identity Management components and a new database for the OracleAS Metadata Repository in an active-passive configuration. In the installer, you select the "Identity Management and OracleAS Metadata Repository" installation type.	Oracle Identity Management components are installed together with the OracleAS Metadata Repository in the same Oracle home in an active-passive configuration.

Table 2–2 (Cont.) Distribution of Components in High Availability Topologies

Topology	OracleAS Metadata Repository	Oracle Identity Management
Distributed OracleAS Cold Failover Cluster (Infrastructure) Topology	Installer installs the Oracle Identity Management components and a new database for the OracleAS Metadata Repository in an active-passive configuration. In the installer, you select the "Identity Management and OracleAS Metadata Repository" installation type.	**Oracle Internet Directory** and **Oracle Directory Integration Platform**: installed together with OracleAS Metadata Repository in active-passive configuration. **OracleAS Single Sign-On** and **Oracle Delegated Administration Services**: installed separately from the other components in an active-active or active-passive configuration.
OracleAS Cold Failover Cluster (Identity Management) Topology	Installed in an existing database.	Installed in an active-passive configuration.
Distributed OracleAS Cold Failover Cluster (Identity Management) Topology	Installed in an existing database.	**Oracle Internet Directory** and **Oracle Directory Integration Platform**: installed in an active-passive configuration. **OracleAS Single Sign-On** and **Oracle Delegated Administration Services**: installed in an active-active or active-passive configuration. The active-active configuration is the more common configuration in this topology.

2.2 Overview of Active-Active Topologies

Oracle Application Server provides an active-active model for its components in OracleAS Clusters topologies. In an OracleAS Clusters, two or more Oracle Application Server instances are configured to serve the same application workload. These instances typically run on different nodes. These nodes do not need to be in a hardware cluster.

You need an external load balancer in front of the nodes. Clients direct requests to these nodes through the load balancer, which then sends the requests to one of the nodes for processing. The load balancer uses its own algorithm to decide which node to send a request to.

You configure the load balancer with a virtual server name and port. When clients need to access an Oracle Identity Management component running on the nodes, they use this virtual server name and port.

In general, the term OracleAS Clusters describes clustering at the Oracle Application Server instance level. However, if it is necessary to call out the specific type of instances being clustered, this document uses the term "OracleAS Clusters (*type*)" to characterize the cluster solution. For example:

- two or more Oracle Identity Management instances are known as OracleAS Cluster (Identity Management)
- two or more J2EE instances are known as OracleAS Clusters (J2EE)

2.2.1 Properties of Active-Active Topologies

Common properties of an OracleAS Clusters topology include:

- Identical instance configuration

The instances are meant to serve the same workload or application. Their identical configuration guarantees that they deliver identical responses to the same request. Note that some configuration properties are allowed to be instance-specific, such as local host name information.

- Managed as a virtual single instance

  Changes in configuration made to one instance usually need to be propagated to the other instances in an active-active topology.

- Independent operation

  The loss of one Oracle Application Server instance in an active-active topology should not affect the ability of the other instances to continue to serve requests.

2.2.2 Advantages of Active-Active Topologies

Advantages of an OracleAS Clusters topology include:

- Increased availability

  An active-active topology has built-in redundancy (multiple Oracle Application Server instances run the same components). Loss of one instance can be tolerated because other instances can continue to serve the same requests.

- Increased scalability and performance

  Multiple identically-configured instances provide the capability to have a distributed workload shared among different machines and processes. New instances can also be added as the demand of the application grows.

2.2.3 High Availability for the OracleAS Metadata Repository in Active-Active Topologies

In OracleAS Cluster (Identity Management) topologies, the Oracle Identity Management components on all nodes are connected to the same directory store database. High availability for this database, which is also used by the OracleAS Metadata Repository, is achieved by using OracleAS RepCA to install the directory store and OracleAS Metadata Repository into an existing database that is already configured for high availability, such as:

- Oracle Real Application Clusters (Oracle RAC) database
- two-node cold failover cluster database

2.3 Overview of Active-Passive Topologies

Oracle Application Server provides an active-passive model for its components using OracleAS Cold Failover Clusters. In an OracleAS Cold Failover Cluster topology, two Oracle Application Server instances are configured to serve the same application workload but only one instance is active at any particular time. These instances run on two different nodes in a hardware cluster. These two nodes also have access to a shared storage, on which you install the Oracle home for the Oracle Application Server instance.

One of the nodes in the hardware cluster is the active node. It mounts the shared storage and runs the Oracle Application Server instance. The other node is the passive, or standby, node. It runs only when the active node fails. During the failover event, the passive node mounts the shared storage and runs the Oracle Application Server instance.

You also need a virtual hostname and virtual IP address to associate with the nodes in the hardware cluster. Clients use this virtual hostname to access the Oracle Application Server components. During normal operation, the virtual hostname and IP address are associated with the active node. During failover, you make the switch: the virtual hostname and IP address are now associated with the passive node.

In general, the term OracleAS Cold Failover Cluster describes clustering at the Oracle Application Server instance level. However, if it is necessary to call out the specific type of instances being clustered, this document will use OracleAS Cold Failover Cluster (*type*) to characterize the cluster solution. For example

- OracleAS Cold Failover Cluster (Identity Management)
- OracleAS Cold Failover Cluster (Middle-Tier)

2.3.1 Properties of Active-Passive Topologies

Common properties of an OracleAS Cold Failover Cluster topology include:

- Shared storage

  The Oracle home for the Oracle Application Server instance is typically installed on storage that is shared by the nodes in the OracleAS Cold Failover Cluster topology. The passive Oracle Application Server instance has access to the same Oracle binaries, configuration files, and data as the active instance.

- Virtual hostname

  During OracleAS Infrastructure installation, you can specify a virtual hostname. This OracleAS Infrastructure virtual hostname can be managed by a hardware cluster or a load balancer and is used by middle-tier and OracleAS Infrastructure components to access the OracleAS Infrastructure.

  The virtual hostname is associated with a virtual IP. This is the name that gives the Oracle Application Server middle tiers a single system view of the OracleAS Infrastructure with the help of a hardware cluster or load balancer. This name-IP entry must be added to the DNS that the site uses, so that the middle-tier nodes can associate with the OracleAS Infrastructure without having to add this entry into their local `/etc/hosts` (or equivalent) file. For example, if the two physical hostnames of the hardware cluster are `node1.mycompany.com` and `node2.mycompany.com`, the single view of this cluster can be provided by the name `selfservice.mycompany.com`. In the DNS, `selfservice` maps to the virtual IP address of the OracleAS Infrastructure, which either floats between `node1` and `node2` via a hardware cluster or maps to node1 and `node2` by a load balancer, all without the middle tier knowing which physical node is active and actually servicing a particular request.

- Failover procedure

  OracleAS Cold Failover Cluster topologies also include a set of scripts and procedures to detect failure of the active instance and to fail over to the passive instance while minimizing downtime.

2.3.2 Advantages of Active-Passive Topologies

Advantages of an OracleAS Cold Failover Cluster topology include:

- Increased availability

  If the active instance fails for any reason or must be taken offline, the passive instance is prepared to take over at any time.

- Reduced operating costs

  In OracleAS Cold Failover Cluster topologies, only one set of Oracle Application Server processes is up and serving requests. Management of the active instance is generally easier than managing an array of active instances.

- Application independence

  Some applications may not be suited to run in an OracleAS Clusters, or active-active, topology. This may include applications that rely heavily on application state or on information stored locally. An active-passive topology has only one instance serving requests at any particular time.

2.3.3 High Availability Options for OracleAS Metadata Repository in Active-Passive Topologies

In OracleAS Cold Failover Cluster topologies, you have the following high availability options for the OracleAS Metadata Repository.

For the OracleAS Cold Failover Cluster (Infrastructure) and distributed OracleAS Cold Failover Cluster (Infrastructure) topologies, the OracleAS Metadata Repository is installed in a new cold failover cluster database.

For the OracleAS Cold Failover Cluster (Identity Management) and distributed OracleAS Cold Failover Cluster (Identity Management), you install the OracleAS Metadata Repository in an existing high availability database such as:

- Oracle RAC database
- cold failover cluster database

2.4 Collocating vs. Distributing Components

In high availability topologies, you can install and run the Oracle Identity Management components from the same Oracle home or you can distribute them to run them on different nodes.

2.4.1 Collocating Oracle Identity Management Components

For the collocated topologies, which are:

- OracleAS Cluster (Identity Management)
- OracleAS Cold Failover Cluster (Infrastructure)
- OracleAS Cold Failover Cluster (Identity Management)

you run the Oracle Identity Management components from the same Oracle home.

Reasons for choosing a collocated topology over a distributed topology include lower cost (you need fewer machines) and ease of management. However, if you need to run some components on nodes that are more secure (located behind additional firewalls), then you need to distribute the Oracle Identity Management components.

2.4.2 Distributing Oracle Identity Management Components

For the distributed topologies:

- Distributed OracleAS Cluster (Identity Management)
- Distributed OracleAS Cold Failover Cluster (Infrastructure)

- Distributed OracleAS Cold Failover Cluster (Identity Management)

you install and run the Oracle Identity Management components on separate nodes. A common distribution model is:

- Oracle Internet Directory and Oracle Directory Integration Platform on one set of nodes
- OracleAS Single Sign-On and Oracle Delegated Administration Services on a different set of nodes

The components are separated in this manner because OracleAS Single Sign-On and Oracle Delegated Administration Services are typically the first components to be accessed by clients and other components. You can run these components on nodes in the DMZ.

For the Oracle Internet Directory and your databases (including the OracleAS Metadata Repository), these components are repositories of data that you want to secure. You should run these components on more secure nodes located behind an additional firewall.

2.4.2.1 Advantages of Distributing Oracle Identity Management Components

Reasons for distributing the Oracle Identity Management components include:

- Security: You might want to run some components, typically the Oracle Internet Directory and database, on nodes that are located behind additional firewalls.
- Performance: You may get better performance by running the components on multiple nodes.
- Choice of high availability topology: You can configure different high availability models for each tier. For example, in the Distributed OracleAS Cold Failover Cluster (Identity Management) Topology, you run OracleAS Single Sign-On and Oracle Delegated Administration Services in an active-active configuration, but run Oracle Internet Directory in an active-passive configuration.
- Performance isolation: You can scale each set of components independently of each other. For example, if the bottleneck is in OracleAS Single Sign-On, you can just increase the number of nodes that are running OracleAS Single Sign-On without changing the number of nodes that are running Oracle Internet Directory.

2.4.2.2 Disadvantages

Multiple installations are required: you need to perform the installations on each node.

You also need to manage, configure, and patch each node separately.

2.5 Choosing the Best High Availability Topology

There is no single best high availability solution for all systems in the world, but there may be a best solution for your system. Perhaps the most important decision in designing a highly available system is choosing the most appropriate high availability architecture or type of redundancy based on service-level requirements needed by a business or application. Understanding the availability requirements of the business is critical since cost is also associated with the different levels of high availability.

Oracle Application Server offers many high availability solutions to meet different service-level requirements. The most comprehensive solution may not necessarily be the best for your business. To choose the correct high availability topology, ensure you understand your business's service-level requirements first.

Here are some questions to determine your high availability needs:

1. Local high availability: does your production system need to be available 24 hours per day, 7 days per week, and 365 days per year?
2. Scalability: is the scalability of multiple active Oracle Application Server instances required?
3. Site-to-site disaster recovery: is this required?

Based on the answers to these questions, you need to make your selection in two dimensions:

1. Instance redundancy: base, active-active, or active-passive.
2. Site-to-site disaster recovery-enabled architecture: yes or no.

Table 2–3 shows the topology choices based on business requirements.

Table 2–3 Service-Level Requirements and Topology Choices

Business Requirements			Topology Choices	
Local High Availability	**Scalability**	**Disaster Recovery**	**Instance Redundancy**	**Disaster Recovery**
N	N	N	Base	N
Y	N	N	Active-passive	N
N	Y	N	Active-active	N
N	N	Y	Base	Y
Y	Y	N	Active-active	N
Y	N	Y	Active-passive	Y
N	Y	Y	Active-active (middle tier) Base (Infrastructure)[1]	Y
Y	Y	Y	Active-active (middle tier) Active-passive and active-active (Infrastructure)[1]	Y

[1] OracleAS Disaster Recovery supports the base, active-passive, and active-active OracleAS Infrastructure architectures. For additional scalability in a base, active-passive, or active-active architecture, extra computing power can be added to the infrastructure hardware (for example, high capacity CPUs, more memory).

Although you can choose different high availability topologies for your Oracle Application Server middle tier and OracleAS Infrastructure, their local high availability and disaster recovery requirements should be identical. Scalability requirements should be evaluated separately for Oracle Application Server middle tier and OracleAS Infrastructure. Typically the OracleAS Infrastructure does not need to be as scalable as the middle tier because it handles fewer identity management requests.

Because of the differences in scalability requirements, deployment choices for the middle tier and the OracleAS Infrastructure may differ in architecture. For example, if your deployment requires local high availability, site-to-site disaster recovery, scalable middle tier but basic OracleAS Infrastructure scalability, you can choose an active-active middle tier, an active-passive OracleAS Infrastructure, and deploy a standby disaster recovery site that mirrors all middle tier and OracleAS Infrastructure configuration in the production site.

3

Active-Active Topologies

This chapter describes the active-active, or OracleAS Cluster (Identity Management), topologies. This chapter contains the following sections:

3.1 Types of Active-Active Topologies

For Oracle Application Server, there are two active-active topologies:

- OracleAS Cluster (Identity Management), shown in Figure 3–1
- Distributed OracleAS Cluster (Identity Management), shown in Figure 3–2

In active-active topologies all the nodes in the topology are active, meaning that they are ready to process requests from clients. An external load balancer directs requests to these nodes. To access the components running on these nodes, clients use the appropriate virtual server name configured on the load balancer. For example, clients trying to access OracleAS Single Sign-On or Oracle Delegated Administration Services use the virtual server name for the HTTP protocol, while clients trying to access Oracle Internet Directory use the virtual server name for the LDAP protocol.

The main difference between the two active-active topologies is that in the distributed topology the OracleAS Single Sign-On and Oracle Delegated Administration Services components run on one set of nodes, and Oracle Internet Directory and Oracle Directory Integration Platform run on a different set of nodes. In the OracleAS Cluster (Identity Management) topology, all these components run on the same set of nodes.

For an overview of these topologies, see Chapter 2, "Overview of Oracle Application Server High Availability Topologies".

Oracle Directory Integration Platform and Replication Server Notes

In active-active topologies, there is only one active instance of Oracle Directory Integration Platform server (`odisrv`) and the replication server (`oidrepld`) running at any given time. In other words, they run on only one node, although you install them on multiple nodes in an active-active topology.

If the node running `odisrv` or `oidrepld` fails, OID Monitor detects the failure and starts them up on another node. See Section 3.7.2, "OID Monitor Details" for details.

OracleAS Certificate Authority Not Supported

OracleAS Certificate Authority is not supported in OracleAS Cluster (Identity Management) topologies. You can install and run OracleAS Certificate Authority separately.

Special Requirements

To run OracleAS Cluster (Identity Management) topologies, you need to meet the following special requirements:

- Load balancers: In order to distribute requests to the nodes, you need a load balancer in front of each set of nodes. The load balancer receives requests from clients and directs each request to a node using a load balancing algorithm.
- Synchronized Time: You need to ensure that the time on all nodes are synchronized using Greenwich Mean Time so that there is a discrepancy of no more than 250 seconds between them.

Installation of the OracleAS Cluster (Identity Management) topologies is covered in the *Oracle Application Server Installation Guide*.

3.1.1 OracleAS Cluster (Identity Management) Topology

This topology, shown in Figure 3–1, consists of two main tiers:

- On one tier, you run the Oracle Identity Management components (Oracle Internet Directory, OracleAS Single Sign-On, Oracle Delegated Administration Services, and Oracle Directory Integration Platform) on two or more nodes. Each node runs the Oracle Identity Management components mentioned above.

  These nodes provide active-active availability for Oracle Identity Management services. OracleAS Single Sign-On and Oracle Delegated Administration Services run on a single `OC4J_SECURITY` instance in each Oracle home. Oracle Internet Directory also runs on each node.

  A load balancer manages traffic to these nodes. To access the components running on these nodes, clients use the appropriate virtual server name configured on the load balancer.

  The nodes running the Oracle Identity Management components should be functionally equivalent.

  OracleAS Single Sign-On and Oracle Delegated Administration Services are configured in an OracleAS Cluster. This means that they are configured identically on all nodes. For example, if you have two nodes, OracleAS Single Sign-On instances running on both nodes have the same configuration, and all Oracle Delegated Administration Services instances have the same configuration. The load balancer can send requests to either instance. If you want to modify the Oracle Delegated Administration Services or OracleAS Single Sign-On configuration, see Section 3.8.5, "Changing Configuration for Components in Active-Active Topologies".

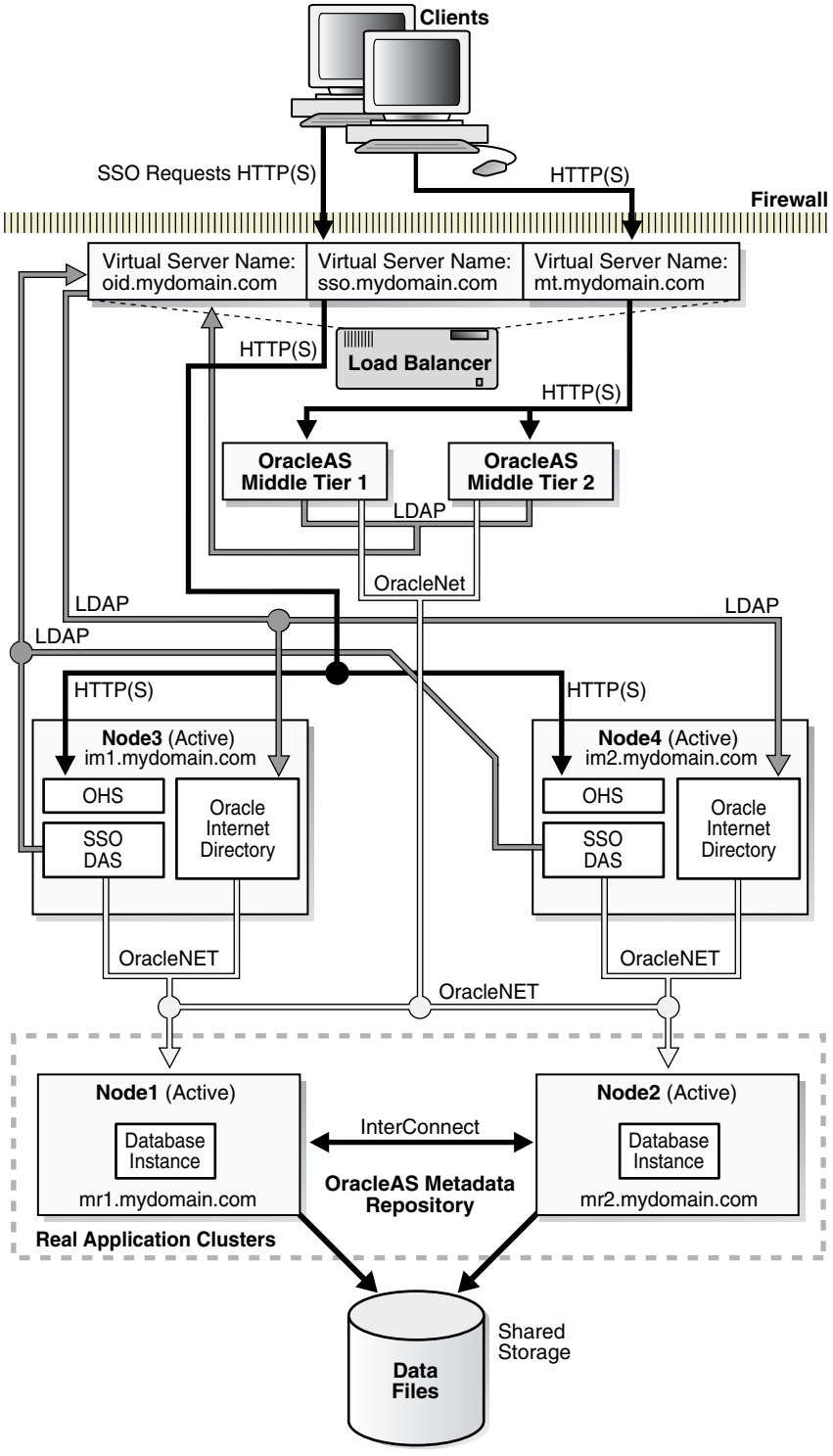
- On another tier, you install and run the OracleAS Metadata Repository on an existing database. This database should already be configured for high availability, such as an Oracle RAC database.

You configure the load balancer with three virtual server names, as shown in Figure 3–1:

- one for OracleAS Single Sign-On and Oracle Delegated Administration Services. Clients use this virtual server name to access these components. For example, in Figure 3–1, this virtual server name is `sso.mydomain.com`.
- one for Oracle Internet Directory. LDAP and JNDI requests from middle tiers and OracleAS Single Sign-On use this virtual server name to access Oracle Internet Directory. For example, in Figure 3–1, this virtual server name is `oid.mydomain.com`.
- one for the middle tiers. Clients use this virtual server name to access the middle tiers. In Figure 3–1, this virtual server name is `mt.mydomain.com`.

OracleAS Single Sign-On, Oracle Delegated Administration Services, and Oracle Internet Directory access the OracleAS Metadata Repository database through Oracle Net load balancing. OracleAS Single Sign-On establishes connection pools to access the database. A connection in the pool can be to any of the database instances in the Oracle RAC system.

Figure 3–1 OracleAS Cluster (Identity Management) Topology



3.1.2 Distributed OracleAS Cluster (Identity Management) Topology

This topology is a variation of the OracleAS Cluster (Identity Management) Topology. Instead of running all the Oracle Identity Management components on each of the

OracleAS Cluster (Identity Management) nodes, you separate out OracleAS Single Sign-On and Oracle Delegated Administration Services to run them on another set of OracleAS Cluster (Identity Management) nodes. Figure 3–2 shows a diagram of the distributed OracleAS Cluster (Identity Management) topology.

The advantage of the distributed topology is that you can deploy the nodes running OracleAS Single Sign-On and Oracle Delegated Administration Services in the DMZ, and deploy the nodes running Oracle Internet Directory inside your intranet, protected by the firewalls, as shown in Figure 3–2.

This topology provides flexibility in placing your components. In this topology:

- You install the OracleAS Metadata Repository in an existing high availability database.
- Oracle Internet Directory and Oracle Directory Integration Platform run on active-active nodes. Typically these nodes are in the same tier as the database.

  These nodes are fronted by a load balancer which directs requests to them. To access Oracle Internet Directory, clients use the LDAP virtual server name configured on the load balancer.
- OracleAS Single Sign-On and Oracle Delegated Administration Services are configured in an OracleAS Cluster. This means that they are configured identically on all nodes. For example, if you have two nodes, OracleAS Single Sign-On instances running on both nodes have the same configuration, and all Oracle Delegated Administration Services instances have the same configuration. The load balancer can send requests to either instance. If you want to modify the Oracle Delegated Administration Services or OracleAS Single Sign-On configuration, see Section 3.8.5, "Changing Configuration for Components in Active-Active Topologies".

  The nodes running OracleAS Single Sign-On and Oracle Delegated Administration Services are active-active nodes. These nodes are placed in the DMZ.

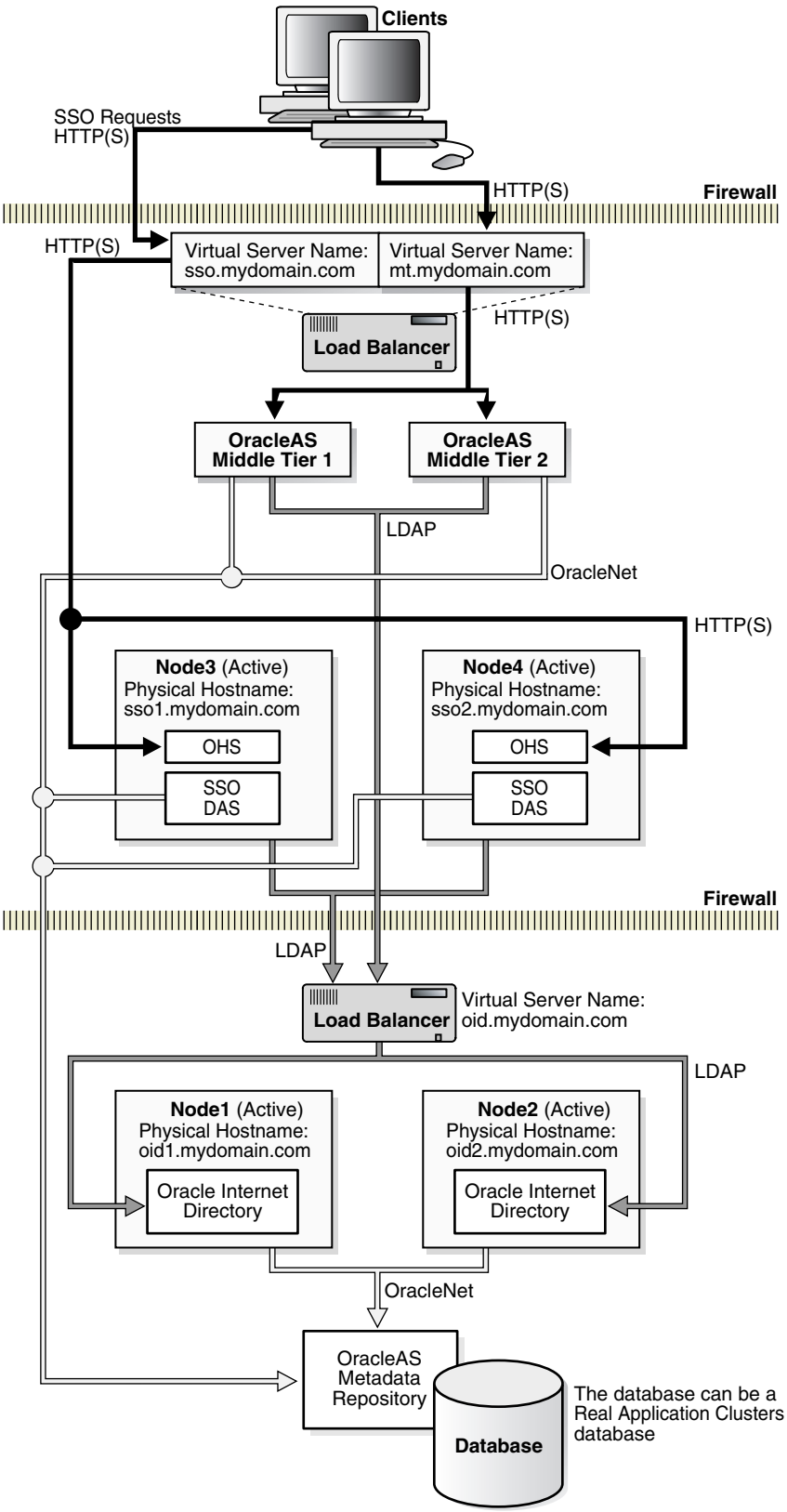
  OracleAS Single Sign-On and Oracle Delegated Administration Services run in the OC4J_SECURITY instance on each node.

  These nodes are also fronted by a load balancer which directs requests to them. Oracle Application Server middle tiers and clients access OracleAS Single Sign-On and Oracle Delegated Administration Services through the HTTP virtual server name configured on this load balancer (`sso.mydomain.com` in Figure 3–2).

Running Middle Tiers on the Same Tier as OracleAS Single Sign-On / Oracle Delegated Administration Services

You can run Oracle Application Server middle tiers on different nodes on the same tier as OracleAS Single Sign-On and Oracle Delegated Administration Services (see Figure 3–2). If there is no firewall separating the middle tier and OracleAS Single Sign-On and Oracle Delegated Administration Services, you can use the same load balancer to load balance the middle tiers also. In this case, you need to configure the load balancer with two virtual server names: one for clients to access OracleAS Single Sign-On and Oracle Delegated Administration Services (`sso.mydomain.com` in Figure 3–2) and one for clients to access the middle tiers (`mt.mydomain.com` in Figure 3–2).

Figure 3–2 Distributed OracleAS Cluster (Identity Management) Topology



3.2 Load Balancer Types and Requirements

In OracleAS Cluster (Identity Management) and distributed OracleAS Cluster (Identity Management) topologies, you need a load balancer to load balance requests among many Oracle Application Server instances.

When several Oracle Application Server instances are grouped to work together, they present themselves as a single virtual entry point to the system, which hides the multiple instance topology from the client. External load balancers can send requests to any Oracle Application Server instance in a cluster, as any instance can service any request.

Active-active topologies are scalable: you can increase the capacity of the system by introducing additional Oracle Application Server instances to the topology. These instances can be installed on separate nodes to allow for redundancy in case of node failure.

Load balancing also improves availability by routing requests to the most available instances. If one instance goes down, or is particularly busy, the external load balancer can send requests to another active instance.

> **Note:** You cannot use a load balancer to load balance requests to Oracle Access Manager servers because these servers use a proprietary protocol.

In Figure 3–1 and Figure 3–2, you can see the location of the load balancer in the topologies.

3.2.1 Load Balancer Types

You can use different types of external load balancers with Oracle Application Server. Table 3–1 describes the different types.

Table 3–1 Types of External Load Balancers

Load Balancer Type	Description
Hardware load balancer	Hardware load balancing involves placing a hardware load balancer in front of a group of Oracle Application Server instances or OracleAS Web Cache. The hardware load balancer routes requests to the Oracle HTTP Server or OracleAS Web Cache instances in a client-transparent fashion.
Software load balancer	Software load balancer involves running some process that intercepts the calls to Oracle Application Server and routes those requests to redundant components.
Lvs network load balancer for Linux	With some Linux operating systems, you can use the operating system to perform network load balancing.
Windows Network Load Balancer (applicable to Windows version of Oracle Application Server)	With some Windows operating systems, you can use the operating system to perform network load balancing. For example, with Microsoft Advanced Server, the NLB functionality enables you to send requests to different machines that share the same virtual IP or MAC address. The servers themselves to do not need to be clustered at the operating system level.

3.2.2 Load Balancer Requirements

The Oracle Identity Management tier uses an external load balancer. This external load balancer should have the following features:

- virtual server name and port configuration

- process failure detection
- persistence configuration for HTTP URLs

Table 3–2 describes these features.

If you are using the same external load balancer for middle tiers, you may need additional features depending on which middle tier components you are running.

Oracle does not provide external load balancers. You can get external load balancers from other companies.

To ensure that your external load balancer can work with Oracle Application Server, check that your external load balancer meets the requirements listed in Table 3–2.

Note that you may not need all the requirements listed in the table. The requirements that you need depend on the topology being considered and on the Oracle Application Server components that are being load balanced.

Table 3–2 External Load Balancer Requirements

External Load Balancer Requirement	Description
Virtual servers and port configuration	A virtual server is a logical address created in a load balancer. The virtual server maps to a group of resources that are load balanced for a request. You need to be able to create virtual server names and ports on your load balancer, and the virtual server names and ports must meet the following requirements: ■ The load balancer should allow configuration of multiple virtual servers. For each virtual server, the load balancer should allow configuration of traffic management on more than one port. For example, for OracleAS Cluster (Identity Management), the load balancer needs to be configured with a virtual server and port for HTTP / HTTPS traffic, and separate virtual servers and ports for LDAP and LDAPS traffic. ■ The virtual server names must be associated with IP addresses and be part of your DNS. Clients must be able to access the external load balancer through the virtual server names.
Persistence (also called "stickiness" by some load balancers)	Persistence refers to the load balancer's ability to establish an identifier for a connection and, based on that identifier, route all subsequent connections from the same client to the same destination host. Some components of Oracle Application Server use persistence in an external load balancer. Here are some examples: ■ For Oracle Internet Directory, do not set a persistence setting for the external load balancer. ■ For OracleAS Single Sign-On, a persistence setting is not required. However, you may set a persistence compatible with Oracle HTTP Server.

Table 3–2 (Cont.) External Load Balancer Requirements

External Load Balancer Requirement	Description
Resource monitoring / port monitoring / process failure detection	You need to set up the external load balancer to detect service and node failures (through notification or some other means) and to stop directing non-Oracle Net traffic to the failed node. If your external load balancer has the ability to automatically detect failures, you should use it. For example, for OracleAS Cluster (Identity Management), specific components that the external load balancer should monitor are Oracle Internet Directory, OracleAS Single Sign-On, and Oracle Delegated Administration Services. To monitor these components, set up monitors for the following protocols: ■ LDAP and LDAPS listen ports ■ HTTP and HTTPS listen ports (depending on the deployment type) These monitors should use the respective protocols to monitor the services. That is, use LDAP for the LDAP port, LDAP over SSL for the LDAP SSL port, and HTTP/HTTPS for the Oracle HTTP Server port. If your external load balancer does not offer these monitors, consult your external load balancer documentation for the best method of setting up the external load balancer to automatically stop routing incoming requests to a service that is unavailable.
Network Address Translation (NAT)	The load balancer should have the capability to perform network address translation (NAT) for traffic being routed from clients to the Oracle Application Server nodes. This is specifically required for OracleAS Portal deployments, where the load balancer should allow enabling NAT for requests originating from within the OracleAS Portal node to the load balancer virtual server (for example, requests such as Parallel Page Engine (PPE) loopbacks and cache invalidation requests).
Fault tolerant mode	It is highly recommended that you configure the load balancer to be in fault-tolerant mode.
Other	It is highly recommended that you configure the load balancer virtual server to return immediately to the calling client when the backend services to which it forwards traffic are unavailable. This is preferred over the client disconnecting on its own after a timeout based on the TCP/IP settings on the client machine because the timeout may be set to a long period of time.

3.3 Installation Highlights

The *Oracle Application Server Installation Guide* provides details on how to install the OracleAS Cluster (Identity Management) and distributed OracleAS Cluster (Identity Management) topologies. Some highlights:

- You need to configure the virtual server names on the load balancers before running the installer.
- For the OracleAS Cluster (Identity Management) topology, you install the components in the following order:
  1. Install OracleAS Metadata Repository on an existing database. This database should be a high availability database, such as an Oracle RAC database or a cold failover cluster database.
  2. Install Oracle Internet Directory, Oracle Directory Integration Platform, OracleAS Single Sign-On, and Oracle Delegated Administration Services on each node. During installation, you enter the LDAP virtual server name and SSL port configured on the external load balancer.

- For the distributed OracleAS Cluster (Identity Management) topology, you install the components in the following order:
  1. Install OracleAS Metadata Repository on an existing database. This database should be a high availability database, such as an Oracle RAC database or a cold failover cluster database.
  2. Install Oracle Internet Directory and Oracle Directory Integration Platform on each node.
  3. Install OracleAS Single Sign-On and Oracle Delegated Administration Services on each node. During installation, you enter the LDAP virtual server name and LDAP SSL port configured on the load balancer.

3.4 LDAP Port Numbers on the Load Balancer and Oracle Internet Directory

In an OracleAS Cluster (Identity Management) topology, the port number used by all the Oracle Internet Directory instances must be the same. You should use the staticports feature during installation to enforce this. For example, you can configure the SSL port to be 636 and the non-SSL port to be 389 for all the Oracle Internet Directory instances in the topology.

The LDAP ports (SSL and non-SSL) configured on the load balancer can be different from the corresponding SSL and non-SSL Oracle Internet Directory ports. However, managing the components might be easier if you use the same port numbers for Oracle Internet Directory and the LDAP ports on the load balancer.

When managing the Oracle Internet Directory instances, using tools such as the Oracle Directory Manager tool or command-line tools, you typically need to specify connect information such as the name of the host running Oracle Internet Directory and the Oracle Internet Directory port number. If you have configured different LDAP port numbers on the load balancer and on Oracle Internet Directory itself, be sure you use the appropriate hostname:port number pair (that is, use the LDAP virtual hostname with the LDAP port configured on the load balancer, or use the physical hostname with the physical port number).

3.5 Backup and Recovery

You can back up and recover files for all the nodes in the OracleAS Cluster (Identity Management) and distributed OracleAS Cluster (Identity Management) topologies using the OracleAS Backup and Recovery Tool. This tool is described in the *Oracle Application Server Administrator's Guide*.

3.6 OracleAS Metadata Repository Tier Details

The nodes on this tier run an Oracle database configured for high availability, such as an Oracle RAC database or a cold failover cluster database. You manage this database as you would any other Oracle database.

If you installed the OracleAS Metadata Repository in an existing Oracle RAC database, node failures are managed by Oracle Net and Oracle RAC. Oracle Net redirects requests to remaining active database instances if any of the other database instances fail.

If you installed the OracleAS Metadata Repository in a cold failover cluster database, node failure is performed by switching the virtual hostname and IP to the standby

node and starting the database processes on that node. Section 7.1.5, "Failing Over a Cold Failover Cluster Database" provides instructions on how to accomplish these tasks.

3.7 Oracle Identity Management Tier Details

This section contains the following topics:

- Section 3.7.1, "Protection Against Process and Node Failures"
- Section 3.7.2, "OID Monitor Details"
- Section 3.7.3, "Oracle Internet Directory Metadata Synchronization"

3.7.1 Protection Against Process and Node Failures

OPMN and the load balancer protect against process failures and node failures.

OPMN runs on each node to provide process management, monitoring, and notification services for OC4J_SECURITY, Oracle HTTP Server, and Oracle Internet Directory processes. If any of these processes fails, OPMN detects the failure and attempts to restart the process.

For managing the Oracle Internet Directory processes, OPMN manages the OID Monitor process ("`oidmon`"), which in turn manages the `oidldapd`, `oidrepld`, and `odisrv` Oracle Internet Directory processes. If `oidldapd`, `oidrepld`, or `odisrv` fails, `oidmon` attempts to restart it locally. Similarly, if `oidmon` fails, OPMN tries to restart it locally.

If the restart is unsuccessful (for example, if OPMN fails to restart `oidmon`, or if `oidmon` fails to restart the Oracle Internet Directory processes), the load balancer detects the failure (usually through a non-response timeout) and directs requests to an active process running on another node in the topology.

Note: Only one `odisrv` process and one `oidrepld` process can be active at any time in an OracleAS Cluster (Identity Management) or distributed OracleAS Cluster (Identity Management) topology. However, you can have multiple `oidldapd` processes running in the same cluster. See the *Oracle Internet Directory Administrator's Guide* for details.

If OC4J_SECURITY is down on a node, the active Oracle HTTP Servers direct traffic to a surviving OC4J_SECURITY instance (this is by virtue of the fact that they are clustered). If Oracle HTTP Server is down on a node, then the surviving Oracle HTTP Servers on the other nodes service the requests.

If a node fails, the load balancer detects the failure and redirects requests to a remaining active node. Because each node provides identical services as the other nodes, all requests can be fulfilled by the remaining active nodes.

Note: If a node goes down or if the processes on a node are brought down due to planned maintenance, you should reconfigure the load balancer not to send traffic to this node.

For information on running Oracle Internet Directory in an OracleAS Cluster (Identity Management) topology, and how directory replication can provide additional high

availability, see Chapter 10, "Deploying Identity Management with Multimaster Replication".

3.7.2 OID Monitor Details

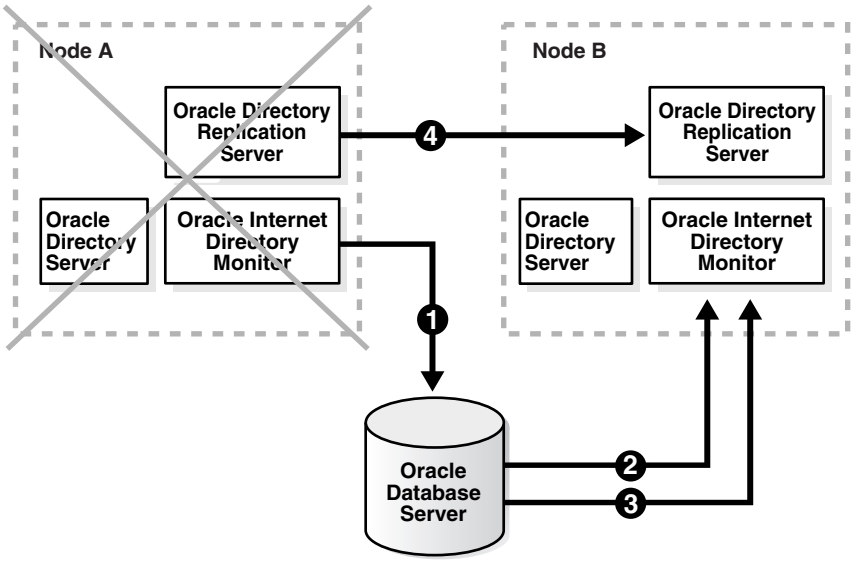
In the OracleAS Cluster (Identity Management) and distributed OracleAS Cluster (Identity Management) topologies, the OID Monitor ("`oidmon`") on each node reports to the other nodes that it is running by sending a message to the Oracle Database every 60 seconds. When it does this, it also polls the database server to verify that all other directory server nodes are also running. If an OID Monitor on one of the nodes has not reported that it is running after a configurable amount of time, the other directory server nodes regard it as having failed. At this point, the following occurs on one of the other nodes that are still running:

1. The OID Monitor on that node brings up the processes that were running on the failed node.
2. The Oracle Directory Integration Platform server (`odisrv`) and the replication server (`oidrepld`) on that node continue processing the operations that were previously underway on the failed node.

   The directory server itself (`oidldapd`) will not be failed over to the other node. This is because `oidldapd` by nature runs in an active-active mode. On the other hand, the Oracle Directory Integration Platform server and the replication server run in an active-passive mode, in the sense that there is only one instance of these servers running at any given time.
3. The OID Monitor on that node logs that it has brought up the processes that were previously running on the failed node.

Figure 3–3 and the accompanying text exemplify this process on two nodes, Node A and Node B.

Figure 3–3 Oracle Internet Directory Failover Process



The Oracle Internet Directory failover process follows these steps (shown in Figure 3–3):

1. Every 60 seconds, the OID Monitor on Node A reports that it is running by sending a message to the database.

2. The OID Monitor on Node B polls the database to learn which, if any, of the other nodes may have failed.
3. If the OID Monitor on Node B learns that Node A has not responded for the configured amount of time (see below for how this is set), it regards Node A as having failed. It then retrieves from the database the necessary information about the Oracle Internet Directory servers that were running on Node A. In this example, it learns that the directory replication server had been running on Node A.

   The failover time is specified in the `orclfailoverenabled` attribute in the DSA config entry ("`cn=dsaconfig, cn=configsets, cn=oracle internet directory`"). By default, the `orclfailoverenabled` attribute is set to `5`; the value is specified in minutes.

   If your deployment requires a longer failover time, then you need to increase the value of the `orclfailoverenabled` attribute.

   If you set the `orclfailoverenabled` attribute to `0`, it means that Oracle Internet Directory processes will not fail over to another node. For example, assume Node A is running a directory replication server and its `orclfailoverenabled` attribute is set to `0`. If Node A fails, the OID Monitor on Node B will not start up the directory replication server on Node B because on Node A, the `orclfailoverenabled` attribute is set to `0`.
4. Because a directory replication server was not already running on Node B, the OID Monitor on Node B starts a directory replication server that corresponds to the directory replication server previously running on Node A.

See Also:

- "Oracle Internet Directory Architecture" in the chapter "Directory Concepts and Architecture" in *Oracle Internet Directory Administrator's Guide* for information about directory server nodes, directory server instances, and the kinds of directory metadata stored in the database
- "Process Control" in the chapter "Directory Administration Tools" in the *Oracle Internet Directory Administrator's Guide*

3.7.2.1 Normal Shutdown vs. Process Failure

Unlike process failures, normal shutdowns are not treated as failovers. After a normal shutdown of Node A, the OID Monitor on Node B does not start the Oracle Internet Directory processes automatically on Node B. Here are two examples to show the differences:

Remember that in OracleAS Cluster (Identity Management) and distributed OracleAS Cluster (Identity Management) topologies, only one host runs the Oracle Directory Integration Platform server process (`odisrv`).

Example of a process failure: Assume Node A runs the directory replication server (`oidrepld`) and/or the Oracle Directory Integration Platform server (`odisrv`). If Node A fails, OID Monitor on Node B starts these processes on Node B after the configured time set in the `orclfailoverenabled` attribute. When Node A is restarted, the OID Monitor on Node A starts the servers automatically and requests the OID Monitor on Node B to stop the servers that were started on Node A.

Note that if the `orclfailoverenabled` attribute is set to `0`, the OID Monitor on Node B will not start up any of the processes that were running on Node A. The `orclfailoverenabled` attribute is described in Section 3.7.2, "OID Monitor Details".

Example of a normal shutdown for `odisrv`: If you stop `odisrv` normally (for example, using the `oidctl` command), it is not considered a failure. This means that the `odisrv` process will not be started up automatically on the other node. You will need to start it up manually using the `oidctl` command. For information on starting `odisrv` through the `oidctl` command, see section 6.3.5, "Configuring an Oracle Directory Integration Platform Server Instance", in the *Oracle Internet Directory Administrator's Guide*.

3.7.2.2 Time Discrepancy Between Nodes

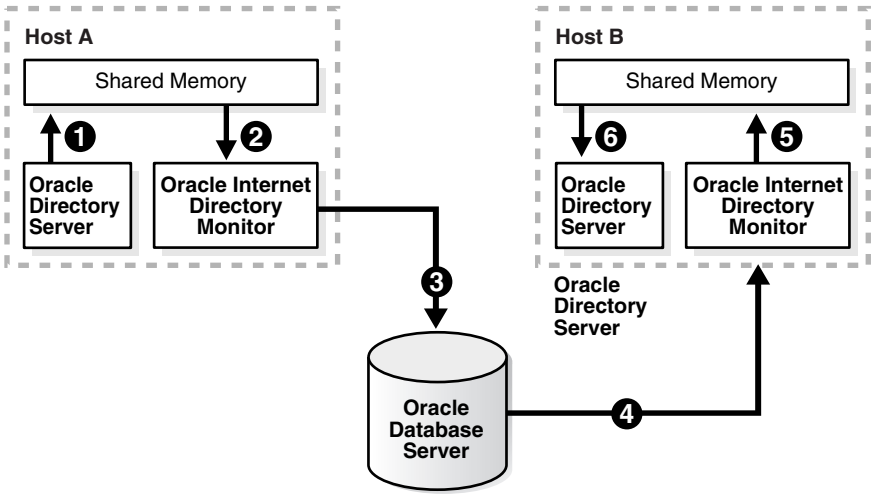
The time on all nodes should be synchronized using Greenwich Mean Time so that there is a discrepancy of no more than 250 seconds between them.

If OID Monitor detects a time discrepancy of more than 250 seconds between the two nodes, the OID Monitor on the node that is behind stops all servers on its node. To correct this problem, synchronize the time on the node that is behind in time. The OID Monitor automatically detects the change in the system time and starts the Oracle Internet Directory servers on its node.

3.7.3 Oracle Internet Directory Metadata Synchronization

In OracleAS Cluster (Identity Management) and distributed OracleAS Cluster (Identity Management) topologies, it is necessary to synchronize Oracle Internet Directory metadata—for example, definitions of object classes, attributes, matching rules, ACPs, and password policies—on all the directory server nodes. Figure 3–4 and the accompanying text exemplify the process in which directory server metadata is synchronized between two directory server nodes, Host A and Host B, in an OracleAS Cluster (Identity Management) topology.

Figure 3–4 Oracle Internet Directory Metadata Synchronization Process



In Figure 3–4, directory server metadata in an OracleAS Cluster (Identity Management) topology is synchronized as follows:

1. On Host A, the directory server writes metadata changes to the shared memory on that same host.
2. OID Monitor on Host A polls the shared memory on that same host. When it discovers a change in the metadata, it retrieves the change.

3. OID Monitor sends the change to the Oracle Database, which is the repository for the directory server metadata in the OracleAS Cluster (Identity Management) environment.
4. OID Monitor on Host B polls the Oracle Database for changes in directory server metadata, and retrieves those changes.
5. OID Monitor on Host B sends the change to the shared memory on that same host.
6. The directory server on Host B polls the shared memory on that same host for metadata changes. It then retrieves and applies those changes.

3.8 Some Useful Procedures

This section describes the following procedures:

- Section 3.8.1, "Using Application Server Control Console"
- Section 3.8.2, "Starting the Components"
- Section 3.8.3, "Stopping the Components"
- Section 3.8.4, "Checking the Status of Oracle Identity Management Components"
- Section 3.8.5, "Changing Configuration for Components in Active-Active Topologies"
- Section 3.8.6, "Changing the Password of the ODS Schema (Used by Oracle Internet Directory)"

3.8.1 Using Application Server Control Console

You can use Application Server Control Console to manage the Oracle Identity Management components in OracleAS Cluster (Identity Management) topologies. To access Application Server Control Console for the OracleAS Cluster (Identity Management) nodes, you use the physical hostname in the Application Server Control Console URL, for example: `http://im1.mydomain.com:1156` (assuming Application Server Control is running on port 1156).

For the OracleAS Metadata Repository, you manage the database using database management tools, such as Oracle Enterprise Manager.

3.8.2 Starting the Components

The steps are slightly different, depending on whether you are running the OracleAS Cluster (Identity Management) or the distributed OracleAS Cluster (Identity Management) topology.

3.8.2.1 For the OracleAS Cluster (Identity Management) Topology

Start up the processes on the different tiers in the following order:

1. Start up the OracleAS Metadata Repository database.
2. On each node in the OracleAS Cluster (Identity Management), run OPMN to start up the Oracle Identity Management components:

   ORACLE_HOME`/opmn/bin/opmnctl startall`

   ORACLE_HOME refers to the Oracle Identity Management's Oracle home.
3. On each node, start up Application Server Control.

```
ORACLE_HOME/bin/emctl start iasconsole
```

3.8.2.2 For the Distributed OracleAS Cluster (Identity Management) Topology

Start up the processes on the different tiers in the following order:

1. Start up the OracleAS Metadata Repository database.
2. On each node in the Oracle Internet Directory tier:
   a. Start up Oracle Internet Directory. You can do this using OPMN.

      ```
      ORACLE_HOME/opmn/bin/opmnctl startall
      ```

      ORACLE_HOME refers to the directory where you installed Oracle Internet Directory.

   b. Start up Application Server Control.

      ```
      ORACLE_HOME/bin/emctl start iasconsole
      ```

3. On each node in the OracleAS Single Sign-On and Oracle Delegated Administration Services tier:
   a. Start up OracleAS Single Sign-On, Oracle Delegated Administration Services, and Oracle HTTP Server. You can do this using OPMN.

      ```
      ORACLE_HOME/opmn/bin/opmnctl startall
      ```

      ORACLE_HOME refers to the OracleAS Single Sign-On / Oracle Delegated Administration Services Oracle home.

   b. Start up Application Server Control.

      ```
      ORACLE_HOME/bin/emctl start iasconsole
      ```

3.8.3 Stopping the Components

The steps are slightly different, depending on whether you are running the OracleAS Cluster (Identity Management) or the distributed OracleAS Cluster (Identity Management) topology.

3.8.3.1 For the OracleAS Cluster (Identity Management) Topology

Stop the processes on the different tiers in the following order:

1. On each node in the OracleAS Cluster (Identity Management), run OPMN to stop the Oracle Identity Management components:

   ```
   ORACLE_HOME/opmn/bin/opmnctl stopall
   ```

   ORACLE_HOME refers to the Oracle Identity Management's Oracle home.

2. Stop the OracleAS Metadata Repository database.
3. Stop Application Server Control.

   ```
   ORACLE_HOME/bin/emctl stop iasconsole
   ```

3.8.3.2 For the Distributed OracleAS Cluster (Identity Management) Topology

Stop the processes on the different tiers in the following order:

1. On each node in the OracleAS Single Sign-On and Oracle Delegated Administration Services tier:
   a. Stop OracleAS Single Sign-On, Oracle Delegated Administration Services, and Oracle HTTP Server. You can do this using OPMN.

      ORACLE_HOME`/opmn/bin/opmnctl stopall`

      ORACLE_HOME refers to the OracleAS Single Sign-On / Oracle Delegated Administration Services Oracle home.

   b. Stop Application Server Control.

      ORACLE_HOME`/bin/emctl stop iasconsole`

2. On each node in the Oracle Internet Directory tier:
   a. Stop Oracle Internet Directory. You can do this using OPMN.

      ORACLE_HOME`/opmn/bin/opmnctl stopall`

      ORACLE_HOME refers to the directory where you installed Oracle Internet Directory.

   b. Stop Application Server Control.

      ORACLE_HOME`/bin/emctl stop iasconsole`

3. Stop the OracleAS Metadata Repository database.

3.8.4 Checking the Status of Oracle Identity Management Components

Use the following commands to check the status of Oracle Identity Management components:

1. Check the status of OPMN and OPMN-managed processes:

   ORACLE_HOME`/opmn/bin/opmnctl status`

2. Check the status of Application Server Control.

   ORACLE_HOME`/bin/emctl status iasconsole`

3. Check the status of Oracle Internet Directory:

   ORACLE_HOME`/ldap/bin/ldapcheck`

   Verify that you can log in to Oracle Internet Directory:

   ORACLE_HOME`/bin/oidadmin`

   Use the following login and password:

   Login: `orcladmin`

   Password: <*orcladmin_password*>

   After installation, the *orcladmin_password* is the same as the ias_admin password.

4. Verify you can log in to OracleAS Single Sign-On:

   `http://`*host*`:`*HTTP_port*`/pls/orasso`

   For *host*, you specify the virtual hostname.

   Login: `orcladmin`

Password: *orcladmin_password*

5. Verify you can log in to Oracle Delegated Administration Services:

   `http://host:HTTP_port/oiddas`

   For *host*, you specify the virtual hostname.

   Login: `orcladmin`

   Password: *orcladmin_password*

3.8.5 Changing Configuration for Components in Active-Active Topologies

The OracleAS Single Sign-On and Oracle Delegated Administration Services instances in active-active topologies need to be configured identically. This means that if you change the configuration for one instance, you also need to make the same change in all the other instances in the topology.

To ensure that configuration stays consistent across the topology, note the following:

- After you make a configuration change to OPMN, Oracle HTTP Server, or OC4J_SECURITY, run the following DCM command to upload the configuration information to the OracleAS Metadata Repository and propagate the change to all instances in the OracleAS Clusters.

  ORACLE_HOME`/dcm/bin/dcmctl updateConfig`

- Configuration changes to Oracle Internet Directory are not automatically managed across the OracleAS Clusters. If you make changes to configuration files, primarily the wallet files, you need to make the same changes manually to all nodes in the OracleAS Clusters.

3.8.6 Changing the Password of the ODS Schema (Used by Oracle Internet Directory)

If you change the password to the Oracle Internet Directory-designated database, then you must update each of the other nodes in the OracleAS Cluster (Identity Management) topology. You can change the ODS database user account password using the `oidpasswd` utility.

To change the ODS database user password, invoke the following command on one of the Oracle Internet Directory nodes:

```
oidpasswd connect=db-conn-str change_oiddb_pwd=true
```

On all other Oracle Internet Directory nodes, invoke the following command to synchronize the password wallet:

```
oidpasswd connect=db-conn-str create_wallet=true
```

> **Note:** If you are running Oracle Internet Directory in an Oracle RAC environment, see Section 9.6, "About Changing the ODS Password on an Oracle RAC System". In an Oracle RAC environment, you have to ensure that the `oidpwdlldap1` file is the same on all your Oracle RAC nodes.

See Also:

- "oidpasswd" in *Oracle Identity Management User Reference* for instructions on how to change the password to the Oracle Application Server-designated database
- "Starting and Stopping Oracle Internet Directory, Replication, and Oracle Directory Integration Platform Servers on a Virtual Host or Cluster Node" in *Oracle Identity Management User Reference*

4

Active-Passive Topologies

This chapter describes the active-passive, or OracleAS Cold Failover Cluster, topologies. This chapter contains the following sections:

4.1 Types of OracleAS Cold Failover Cluster Topologies

With Oracle Application Server, you can create the following OracleAS Cold Failover Cluster topologies:

- OracleAS Cold Failover Cluster (Infrastructure) Topology
- Distributed OracleAS Cold Failover Cluster (Infrastructure) Topology
- OracleAS Cold Failover Cluster (Identity Management) Topology
- Distributed OracleAS Cold Failover Cluster (Identity Management) Topology

OracleAS Cold Failover Cluster topologies are active-passive topologies, which mean that only you have one active node that is running the Oracle Application Server components, and another node, the passive node, on standby. If the active node fails, the passive node takes over and runs the Oracle Application Server components.

As with OracleAS Clusters, or active-active, topologies, OracleAS Cold Failover Cluster topologies protect against process failures and node failures. Section 4.5, "Protection Against Process Failures and Node Failures" provides the details.

In the distributed version of the OracleAS Cold Failover Cluster topologies, the Oracle Identity Management components run on different sets of nodes, as follows: the OracleAS Single Sign-On and Oracle Delegated Administration Services components run on one set of nodes, while the Oracle Internet Directory and Oracle Directory Integration Platform components run on another set of nodes.

Also, in the distributed topologies, OracleAS Single Sign-On/Oracle Delegated Administration Services can be installed in either an active-active topology or an active-passive topology.

In the non-distributed version of the topologies, all the Oracle Identity Management components run on the same node.

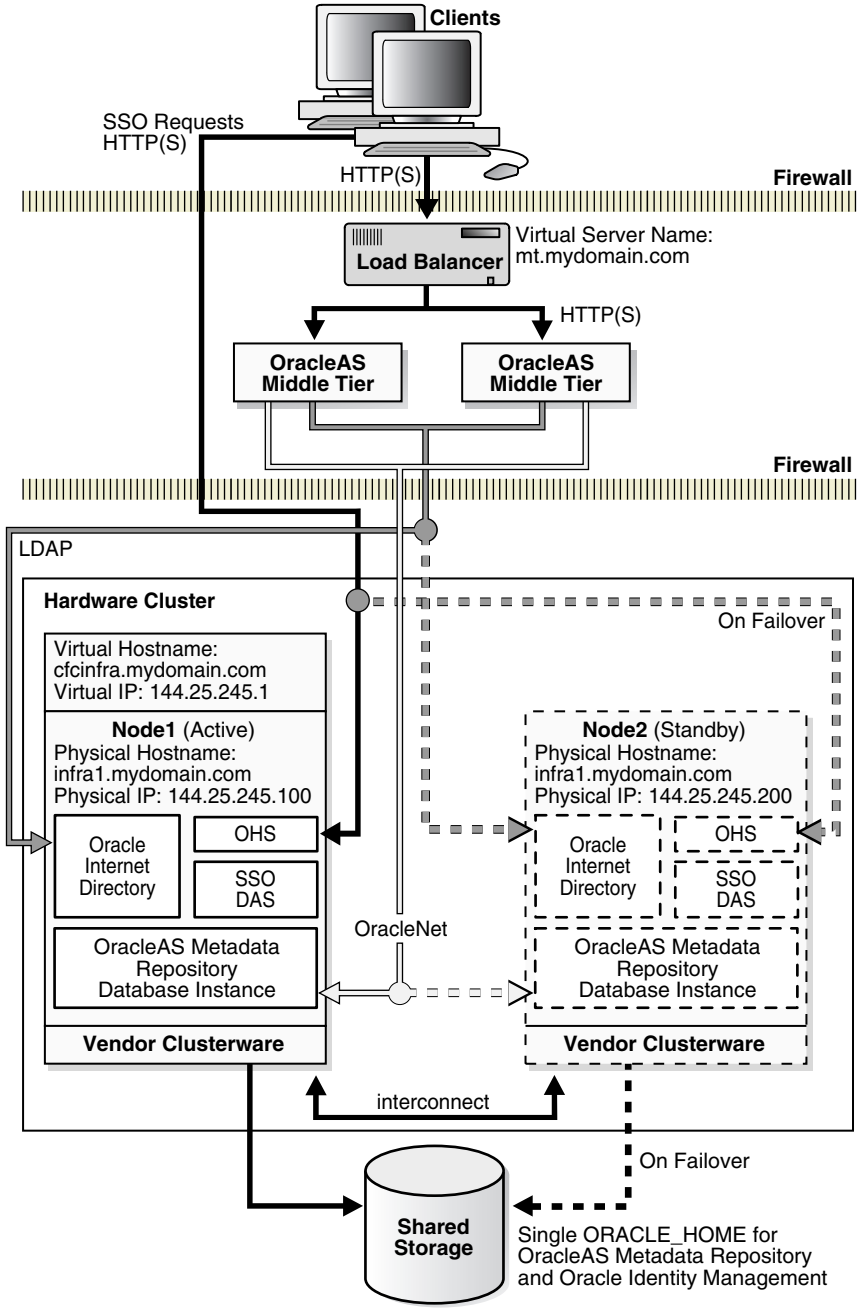
The following sections describe the topologies in detail.

4.1.1 OracleAS Cold Failover Cluster (Infrastructure) Topology

Figure 4–1 shows an OracleAS Cold Failover Cluster (Infrastructure). It consists of:

- two nodes in a hardware cluster. One of the nodes is the active node, and the other node is the passive node. The active node runs all the components and responds to all requests. If the active node goes down for any reason, the passive node takes over: it runs the components and handles all the requests.
- shared storage that can be mounted by both nodes. The shared storage contains the single Oracle home for the Oracle Identity Management and OracleAS Metadata Repository. Only one node, the active node, is mounted to the shared storage at any time.
- virtual hostname and virtual IP address. The virtual hostname and virtual IP address point to the active node. Because clients use this virtual hostname, instead of the node's physical hostname, to access the OracleAS Infrastructure, clients do not need to know which node in the cluster is running the OracleAS Infrastructure components. You specify this virtual hostname during installation.
- On Windows, you also need to install Oracle Fail Safe on the local storage of each of the OracleAS Cold Failover Cluster (Infrastructure) nodes.

Figure 4–1 OracleAS Cold Failover Cluster (Infrastructure): Normal Operation



4.1.2 Distributed OracleAS Cold Failover Cluster (Infrastructure) Topology

In a distributed OracleAS Cold Failover Cluster (Infrastructure) topology (see Figure 4–2), you distribute the Oracle Identity Management components to create a more secure environment. You deploy OracleAS Single Sign-On and Oracle Delegated Administration Services separately from the other OracleAS Infrastructure components.

Typically, you run OracleAS Single Sign-On and Oracle Delegated Administration Services between two firewalls (in a DMZ), and Oracle Internet Directory, Oracle

Directory Integration Platform, and OracleAS Metadata Repository behind the inner firewall, as shown in Figure 4–2.

Clients from outside the first firewall can access OracleAS Single Sign-On and Oracle Delegated Administration Services using the virtual hostname (`sso.mydomain.com` in Figure 4–2). The second firewall prevents clients from accessing the OracleAS Metadata Repository and Oracle Internet Directory directly.

To access Oracle Internet Directory, OracleAS Single Sign-On and Oracle Delegated Administration Services use the virtual hostname (`oidmr.mydomain.com` in Figure 4–2).

The figure also shows Oracle Application Server middle tiers running on the same nodes as OracleAS Single Sign-On and Oracle Delegated Administration Services.

Table 4–1 lists the tiers in this topology:

Table 4–1 Tiers in a Distributed OracleAS Cold Failover Cluster (Infrastructure)

Tier	Configuration
OracleAS Metadata Repository, Oracle Internet Directory, Oracle Directory Integration Platform	Active-passive
OracleAS Single Sign-On and Oracle Delegated Administration Services	Active-active or active-passive

If you want to run OracleAS Single Sign-On and Oracle Delegated Administration Services in active-active configuration instead of active-passive configuration, you will need a load balancer to distribute requests among the active-active instances. This is similar to the OracleAS Single Sign-On and Oracle Delegated Administration Services tier described in Section 3.1.2, "Distributed OracleAS Cluster (Identity Management) Topology".

On Windows, you have to install Oracle Fail Safe on the local storage of each of the nodes in active-passive configuration, that is, the nodes that are running OracleAS Metadata Repository and Oracle Internet Directory. If you are running OracleAS Single Sign-On and Oracle Delegated Administration Services in active-passive configuration, then these nodes also need to have Oracle Fail Safe.

Only one node of the hardware cluster is active at any time. The virtual hostname (`oidmr.mydomain.com`) points to the active node. The shared storage is mounted only on the active node.

To access the components running in this tier, clients use the virtual hostname. For example, to access the OracleAS Metadata Repository or Oracle Internet Directory, you use the virtual hostname (`oidmr.mydomain.com`).

OPMN runs on this tier to manage the Oracle Internet Directory processes.

If the active tier fails, the clusterware fails over the OracleAS Metadata Repository, Oracle Internet Directory, Application Server Control, and OPMN processes to the standby node.

The clusterware also mounts the shared storage on the new active node and associates the virtual hostname and virtual IP address with the new active node.

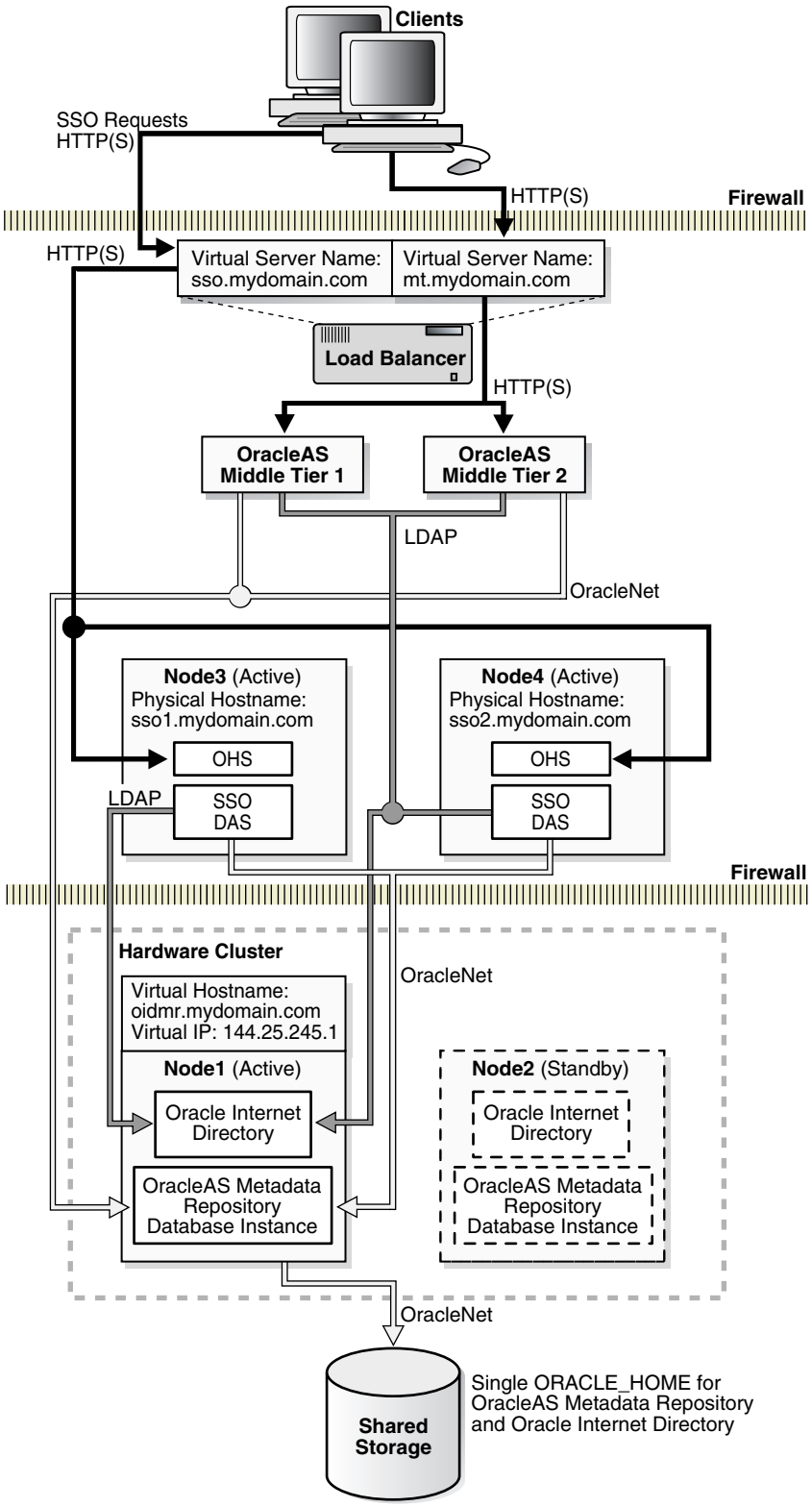
Failover from the active node to the passive node occurs at the node level. All the components running on the active node (Oracle Internet Directory, Oracle Directory Integration Platform, and OracleAS Metadata Repository) fail over together to the passive node.

On Windows, Oracle Fail Safe performs the failover.

Typically, OPMN monitors the Oracle Application Server processes for you, so you do not have to monitor them yourself. However, if you want to monitor them manually, you can use Application Server Control or commands to monitor the status of OracleAS Infrastructure components running on all the nodes.

See Section 7.4, "Checking the Status of OracleAS Metadata Repository" and Section 3.8.4, "Checking the Status of Oracle Identity Management Components" for a list of commands that you can run. Make sure you run the commands on nodes in the appropriate tier.

Figure 4–2 Distributed OracleAS Cold Failover Cluster (Infrastructure) Topology



4.1.3 OracleAS Cold Failover Cluster (Identity Management) Topology

In an OracleAS Cold Failover Cluster (Identity Management) topology (see Figure 4–3), you install and run the Oracle Identity Management components in an OracleAS Cold Failover Cluster. For the OracleAS Metadata Repository, you install it in an existing database using the OracleAS RepCA. This database should be a high availability database, such as an Oracle Real Application Clusters (Oracle RAC) database or a cold failover cluster database. Figure 4–3 shows a cold failover cluster database in the topology.

Table 4–2 lists the tiers in this topology:

Table 4–2 Tiers in an OracleAS Cold Failover Cluster (Identity Management)

Tier	Configuration
OracleAS Metadata Repository	Installed in an existing database
Oracle Identity Management components	Active-passive

The nodes running the Oracle Identity Management components need to be in a hardware cluster. You also need a shared storage for these hardware cluster nodes. You will install the Oracle home for the Oracle Identity Management components on the shared storage.

You need a virtual hostname and virtual IP address for the hardware cluster nodes running the Oracle Identity Management components. Clients, including middle tiers, use the virtual hostname to access the Oracle Identity Management components.

On Windows, you have to install Oracle Fail Safe on the local storage of each node running Oracle Identity Management.

Middle-tier components and applications access Oracle Identity Management services by making LDAP requests to Oracle Internet Directory, and HTTP/HTTPS requests to OracleAS Single Sign-On and Oracle Delegated Administration Services.

Clients can perform single sign-on by making direct HTTP/HTTPS requests to OracleAS Single Sign-On server using the single sign-on URL. This URL uses the virtual hostname configured for Oracle Identity Management.

OracleAS Single Sign-On establishes connection pools to access the OracleAS Metadata Repository database. A connection in the pool uses Oracle Net to communicate with the active database instance(s). Oracle Net is also used by middle-tier components and Oracle Internet Directory to connect to the database.

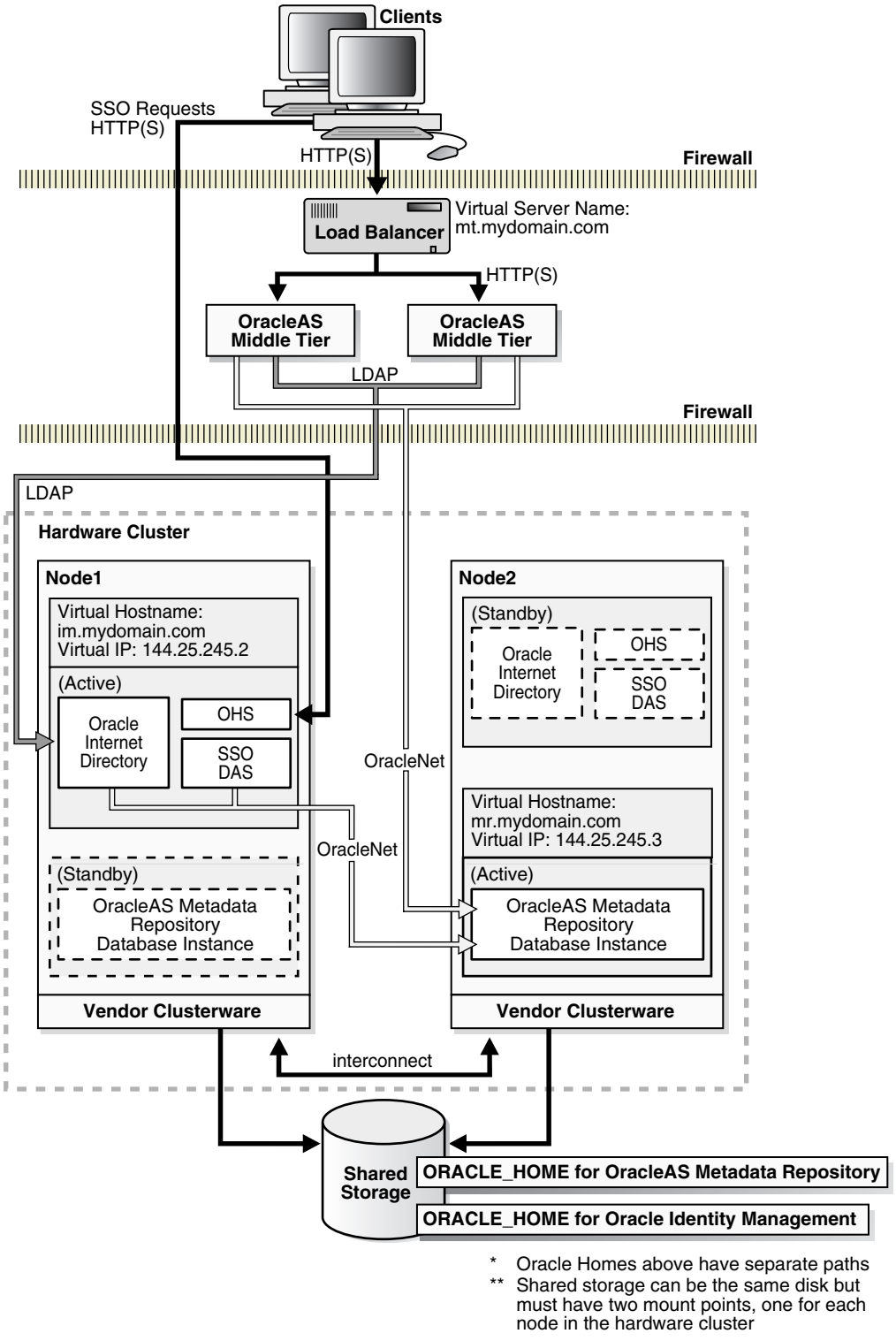
If the node on which the Oracle Identity Management components are running fails, the components fail over to the other node. The virtual hostname and IP also switch to point to the new active node.

If You Are Using a Cold Failover Cluster Database

If you are using a cold failover cluster database, you can install and run Oracle Identity Management on the same nodes as the database. You can also make each node an active node: one node can be the active node for the Oracle Identity Management components, and the other node can be the active node for the database. For example, in Figure 4–3, Node 1 is the active node for Oracle Identity Management, but Node 2 is the active node for the database containing the OracleAS Metadata Repository. This enables you to use both nodes at the same time. If one node fails, then processes running on that node are failed over to the other node, and that node runs all the processes (database and Oracle Identity Management).

To do this, you need to set up separate virtual hostnames and virtual IP addresses for Oracle Identity Management and the database because they point to different active nodes. In Figure 4–3, the virtual hostname for Oracle Identity Management, `im.mydomain.com`, points to Node 1, but the virtual hostname for the database, `mr.mydomain.com`, points to Node 2.

Figure 4–3 OracleAS Cold Failover Cluster (Identity Management) Topology



4.1.4 Distributed OracleAS Cold Failover Cluster (Identity Management) Topology

This topology is similar to the one described in Section 4.1.3, "OracleAS Cold Failover Cluster (Identity Management) Topology" except that you install and run OracleAS Single Sign-On and Oracle Delegated Administration Services on one set of nodes, and Oracle Internet Directory on another set of nodes.

You run the OracleAS Single Sign-On and Oracle Delegated Administration Services nodes in an active-active configuration, which means that you place a load balancer in front of these nodes. The load balancer directs requests to these nodes.

For the Oracle Internet Directory nodes, you run them in an active-passive configuration. If you have an existing cold failover cluster database, you can install Oracle Internet Directory on the same nodes as the database.

For the OracleAS Metadata Repository, you can install it, using OracleAS RepCA, in an existing Oracle RAC database for active-active availability or in a cold failover cluster database for active-passive availability.

You might choose this topology to create a more secure configuration. This topology enables you to run OracleAS Single Sign-On and Oracle Delegated Administration Services in the DMZ, and Oracle Internet Directory and the OracleAS Metadata Repository database in your intranet behind the DMZ.

Figure 4–4 shows a distributed OracleAS Cold Failover Cluster (Identity Management). The OracleAS Metadata Repository is installed in an existing cold failover cluster database. Oracle Internet Directory is installed on the same nodes as the cold failover cluster database.

If you install Oracle Internet Directory on the same cluster as the cold failover database, you need separate virtual hostnames and virtual IP addresses for the database and for Oracle Internet Directory.

Table 4–3 lists the tiers in this topology:

Table 4–3 Tiers in a Distributed OracleAS Cold Failover Cluster (Identity Management)

Tier	Configuration
OracleAS Metadata Repository	Installed in an existing database
Oracle Internet Directory and Oracle Directory Integration Platform	Active-passive
OracleAS Single Sign-On and Oracle Delegated Administration Services	Active-passive or active-active

On Windows, you also have to install Oracle Fail Safe on the local storage of each active-passive node.

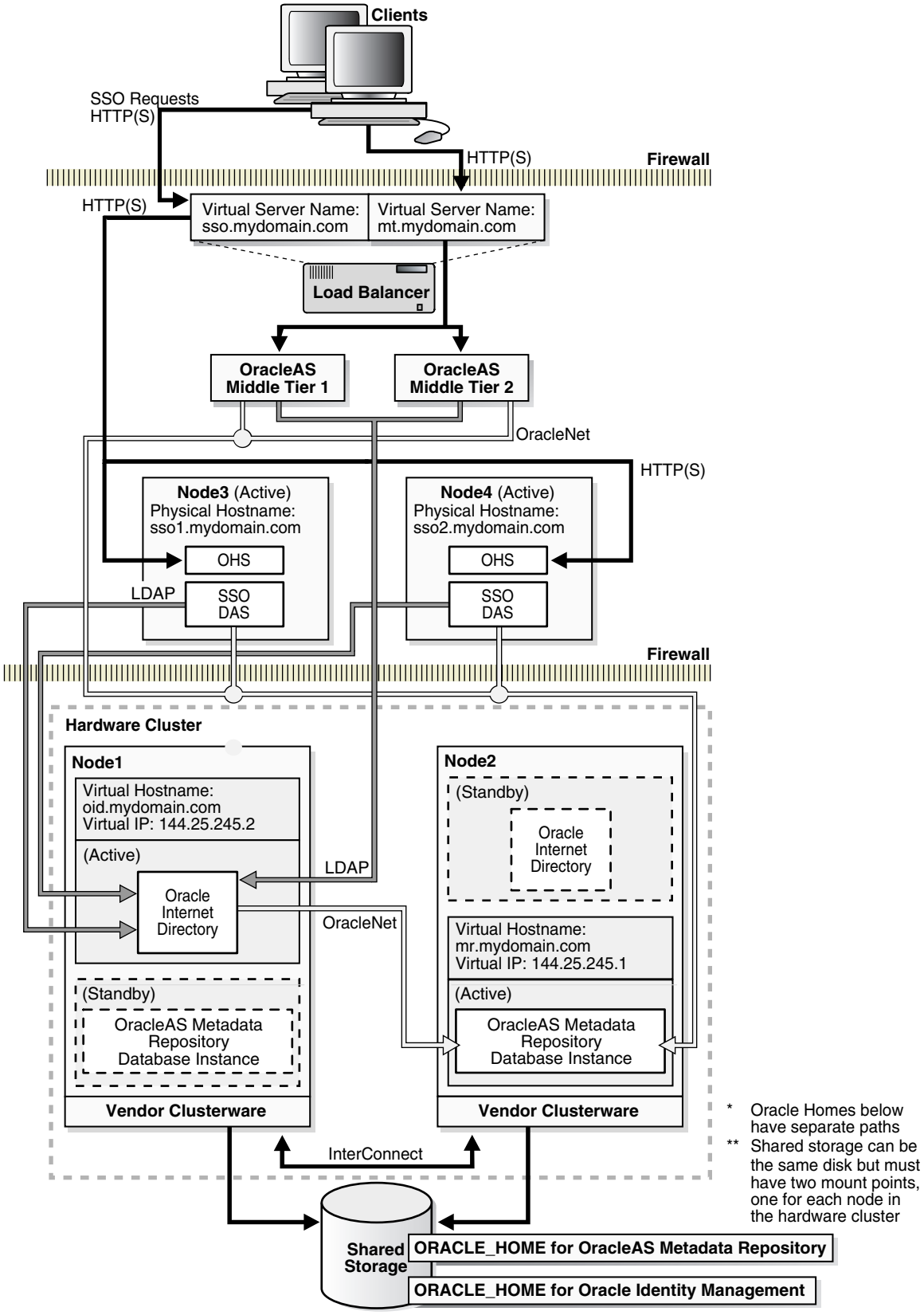
If You Are Running a Cold Failover Cluster Database

If you are using a cold failover cluster database, you can install and run Oracle Internet Directory on the same hardware cluster nodes as the database. You can also make each node an active node: one node can be the active node for Oracle Internet Directory, and the other node can be the active node for the database. For example, in Figure 4–4, Node 1 is the active node for Oracle Internet Directory, but Node 2 is the active node for the database containing the OracleAS Metadata Repository. This enables you to use both nodes at the same time. If one node fails, then processes running on that node are failed over to the other node, and that node runs all the processes (database and Oracle Internet Directory).

To do this, you need to set up separate virtual hostnames and virtual IP addresses for Oracle Internet Directory and the database because they point to different active

nodes. In Figure 4–4, the virtual hostname for Oracle Internet Directory, `oid.mydomain.com`, points to Node 1, but the virtual hostname for the database, `mr.mydomain.com`, points to Node 2.

Figure 4–4 Distributed OracleAS Cold Failover Cluster (Identity Management)



4.2 Common Characteristics of OracleAS Cold Failover Cluster Topologies

OracleAS Cold Failover Cluster topologies run Oracle Application Server components in active-passive mode, but in some topologies you may run some of the components in active-active mode.

OracleAS Cold Failover Cluster topologies have the following characteristics in common:

- Each topology includes two nodes in a hardware cluster. One of these nodes is active, and the other node is passive. The active node runs the Oracle Application Server components and handles all the requests from clients. If the active node goes down for any reason, the passive node takes over and runs the components.

  In the distributed version of the OracleAS Cold Failover Cluster topologies, you may have multiple tiers of hardware cluster nodes running the Oracle Application Server components. For example, in the distributed OracleAS Cold Failover Cluster (Identity Management) topology (see Figure 4–4), one set of hardware cluster nodes runs OracleAS Single Sign-On and Oracle Delegated Administration Services, while another set of hardware cluster nodes runs Oracle Internet Directory and Oracle Directory Integration Platform.

- Each set of hardware cluster nodes is associated with shared storage. You install the Oracle home for Oracle Application Server on the shared storage.

  The active node mounts this shared storage during normal operation. If the active node goes down, then the passive node mounts the shared storage.

- Each set of hardware cluster nodes is also associated with a virtual hostname and virtual IP address. The virtual hostname and virtual IP address point to the active node. Clients use this virtual hostname, instead of the node's physical hostname, to access the components running on the hardware cluster. Clients do not need to know which node in the hardware cluster is actually processing their requests.

  The example topology in Figure 4–5 uses the virtual hostname `cfcinfra.mydomain.com` and virtual IP 144.25.245.1 are used. When node 1 fails, a failover event occurs, and the virtual hostname and IP are moved to the standby node (node 2), which becomes the active node. The failure of the active node is transparent to the clients.

4.2.1 OracleAS Cold Failover Cluster Topologies on Microsoft Windows

On Microsoft Windows, the OracleAS Cold Failover Cluster topologies have the characteristics described in the previous section, but it also has these unique features:

- The nodes in the hardware cluster need to have Microsoft Cluster Server software for managing high availability for the cluster.
- You need to install Oracle Fail Safe on the local storage of each node. Oracle Fail Safe works with Microsoft Cluster Server to manage the following:
  - virtual hostname and IP address
  - OracleAS Metadata Repository database
  - OPMN
  - Application Server Control Console
  - Database Console

The integration of Oracle Fail Safe and Microsoft Cluster Server provides an easy-to-manage environment and automatic failover functionality in the OracleAS Cold Failover Cluster topologies. The OracleAS Metadata Repository database, its TNS listener, and OPMN run as Windows services and are monitored by Oracle Fail Safe and Microsoft Cluster Server. If any of these services fails, Microsoft Cluster Server tries to restart the service three times (the default setting) before failing the group to the standby node. Additionally, OPMN monitors, starts, and restarts the Oracle Internet Directory, OC4J, and Oracle HTTP Server components.

Resource Groups

Central to the Windows OracleAS Cold Failover Cluster topologies is the concept of resource groups. A group is a collection of resources that you set up in Oracle Fail Safe. During failover from the active node to the standby node, the resources associated with the group fail over as a unit. When you install an OracleAS Cold Failover Cluster topology, you create a group for the topology.

The resources in a group depend on the Oracle Application Server components you are running on the hardware cluster nodes. For the OracleAS Cold Failover Cluster (Infrastructure) topology, where the hardware cluster nodes run all the OracleAS Infrastructure components, the group consists of the following:

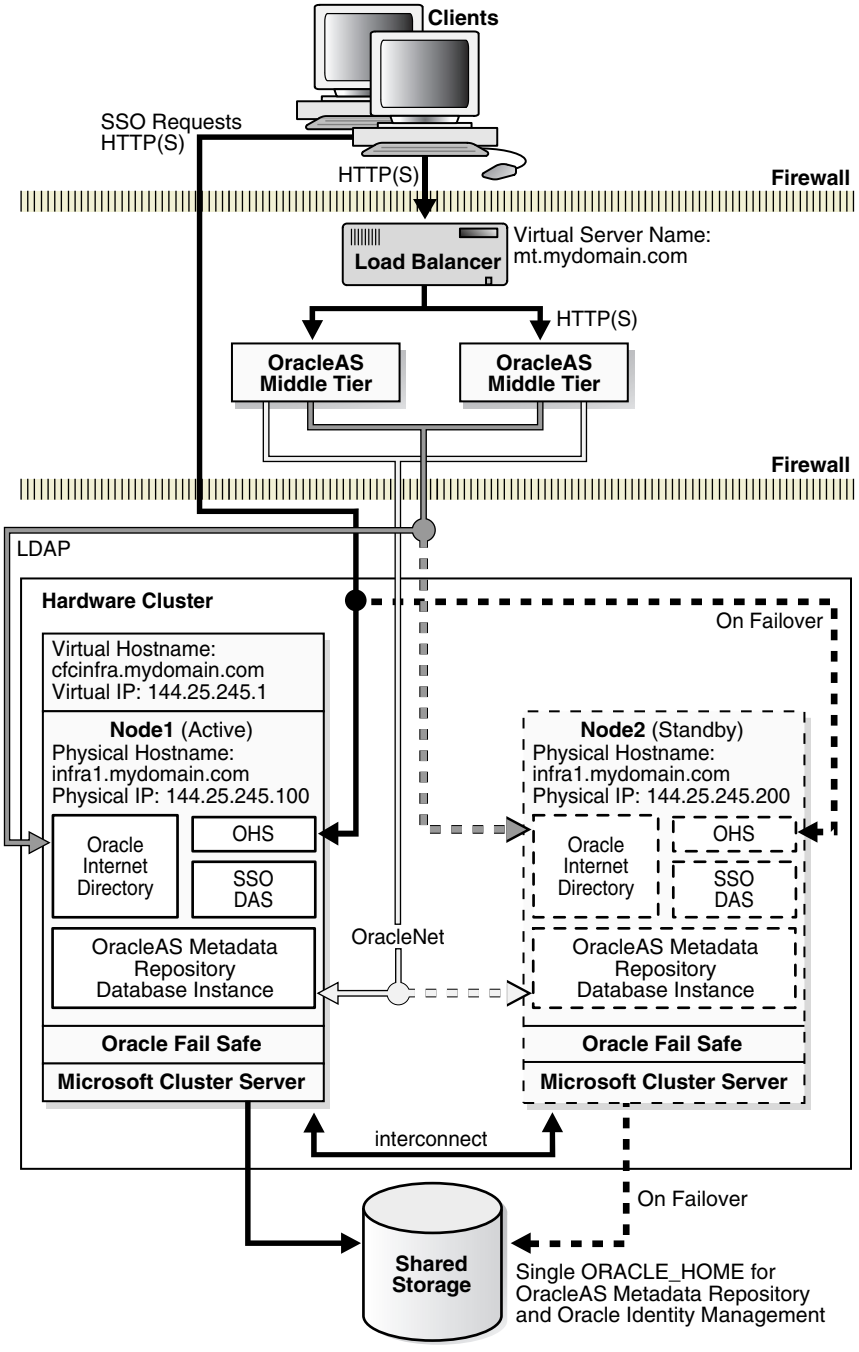
- virtual hostname and virtual IP address for the cluster
- shared storage
- OracleAS Metadata Repository database
- TNS listener for the database
- OPMN
- Application Server Control Console
- Database Console

In other topologies, the resource group contains a subset of the components. For example, on hardware cluster nodes that run only Oracle Internet Directory and Oracle Directory Integration Platform, the resource group contains only the following:

- virtual hostname and virtual IP address for the hardware cluster
- shared storage
- OPMN
- Application Server Control Console

> **Note:** Only static IP addresses can be used in the OracleAS Cold Failover Cluster (Infrastructure) topology for Windows.

Figure 4–5 OracleAS Cold Failover Cluster (Infrastructure) on Microsoft Windows



4.3 Backup and Recovery Procedure

For backing up OracleAS Cold Failover Cluster environments and recovering these backups during failures, use the backup and recovery procedures provided in the *Oracle Application Server Administrator's Guide*.

Additionally, the following considerations should be noted:

Backup considerations:

- Oracle recommends that you place archive logs for the OracleAS Metadata Repository on the shared disk. This ensures that, when failing over from one cluster node to another in the case of media recovery, the archive logs are also failed over and available.
- You can generate archive logs to a local file system; however, the same path must be available during runtime on whichever node is hosting the OracleAS Infrastructure instance.
- Proper capacity planning is required in order to ensure adequate space is available to store the desired number of archive logs.

Recovery considerations:

- If archive logs are stored on a local file system, in the case of media recovery, all archive logs must be made available to the application server instance performing the recovery. Recovery can be performed on either node of the cluster.

4.4 OracleAS Metadata Repository Tier Details

In the OracleAS Cold Failover Cluster (Infrastructure) and distributed OracleAS Cold Failover Cluster (Infrastructure) topologies, the OracleAS Metadata Repository and Oracle Identity Management components run on the same tier. The OracleAS Metadata Repository is installed by the installer in a cold failover cluster database.

In the other topologies, OracleAS Cold Failover Cluster (Identity Management) and distributed OracleAS Cold Failover Cluster (Identity Management), the OracleAS Metadata Repository is installed separately from Oracle Identity Management. Typically, you would install the OracleAS Metadata Repository in an existing high availability database, such as an Oracle RAC database.

4.4.1 Using Database Console to Manage the Cold Failover Cluster Database

If you are running the OracleAS Metadata Repository on a cold failover cluster database, you can use the database console to manage it. Note that before starting or stopping the database console, you need to set the ORACLE_HOSTNAME environment variable to the virtual hostname of the hardware cluster. For example, in the topology shown in Figure 4–2, you would set the ORACLE_HOSTNAME environment variable to `oidmr.mydomain.com`.

On UNIX, you can set the environment variable as follows:

C shell example:

```
> setenv ORACLE_HOSTNAME oidmr.mydomain.com
```

Bourne or Korn shell example:

```
> ORACLE_HOSTNAME=oidmr.mydomain.com
> export ORACLE_HOSTNAME
```

On Windows, you can set the environment variable using the System control panel. Select the Advanced tab to access the environment variable section.

After setting ORACLE_HOSTNAME, you can start or stop the database console (you also need to set the ORACLE_HOME and ORACLE_SID environment variables before starting the database console):

```
> cd ORACLE_HOME/bin
```

```
> emctl start dbconsole
```

4.4.2 Using Automatic Storage Management (ASM)

If you are using the Automatic Storage Management (ASM) feature of Oracle Database 10*g* for the OracleAS Metadata Repository, you must configure the Cluster Synchronization Services (CSS) daemon on the standby node. The CSS daemon synchronizes ASM instances with the database instances that use the ASM instances for database file storage. Specific instructions are provided in the OracleAS Cold Failover Cluster chapter in the *Oracle Application Server Installation Guide*.

4.5 Protection Against Process Failures and Node Failures

OPMN and the hardware cluster protect against process failures and node failures.

To protect against process failures, OPMN runs on the active node to provide process management, monitoring, and notification services for the `OC4J_SECURITY`, Oracle HTTP Server, and Oracle Internet Directory processes. If any of these processes fails, OPMN detects the failure and attempts to restart it. If the restart is unsuccessful, the clusterware detects the failure and fails over all the processes to the passive node.

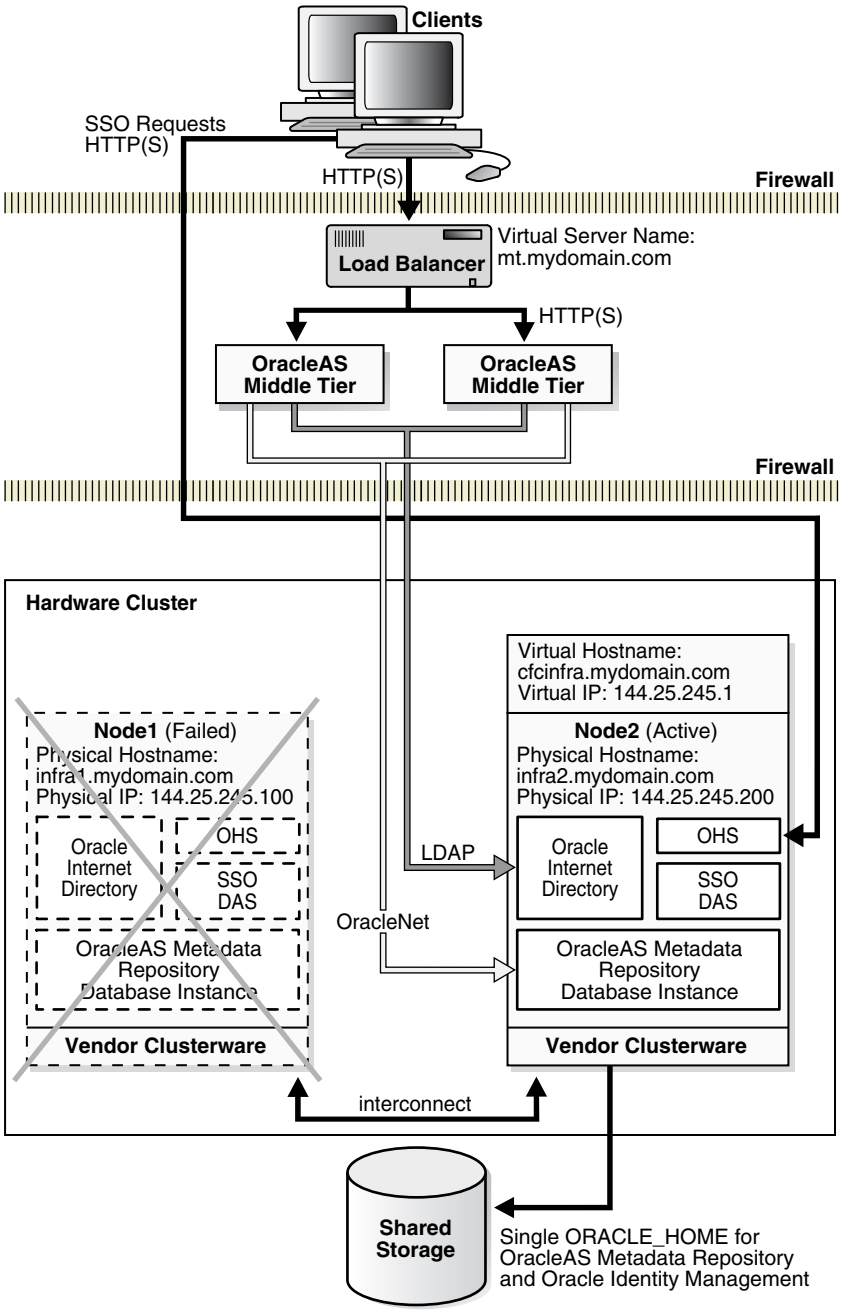
To manage Oracle Internet Directory, OPMN monitors the OID Monitor process ("`oidmon`"), which in turn monitors the `oidldapd`, `oidrepld`, and `odisrv` Oracle Internet Directory processes. If `oidldapd`, `oidrepld`, or `odisrv` fails, `oidmon` attempts to restart it locally. Similarly, if `oidmon` fails, OPMN tries to restart it locally.

If a node fails, the clusterware detects the failure and fails over all the processes to the passive node.

Note: While the hardware cluster framework can start, monitor, or fail over OracleAS Infrastructure processes, these actions are not automatic. You have to do them manually, create scripts to automate them, or use scripts provided by the cluster vendor for OracleAS Cold Failover Cluster.

If the primary node fails, the virtual IP address is manually enabled on the secondary node (Figure 4–6). All the OracleAS Infrastructure processes are then started on the secondary node. Middle tiers accessing the OracleAS Infrastructure will see a temporary loss of service as the virtual IP and the shared storage are moved over to Node2, and the database, database listener, and all other OracleAS Infrastructure processes are started. Once the processes are up, middle-tier processes that were retrying during this time are reconnected. New connections are not aware that a failover has occurred.

Figure 4–6 OracleAS Cold Failover Cluster (Infrastructure): After Failover



4.5.1 Failover on Windows Systems

Figure 4–7, Figure 4–8, and Figure 4–9 show the Oracle Fail Safe Manager screens for a failover operation from the active node to the standby node on Windows.

Figure 4–7 Screen 1 Performing Failover for Oracle Identity Management in an Active-Passive Configuration

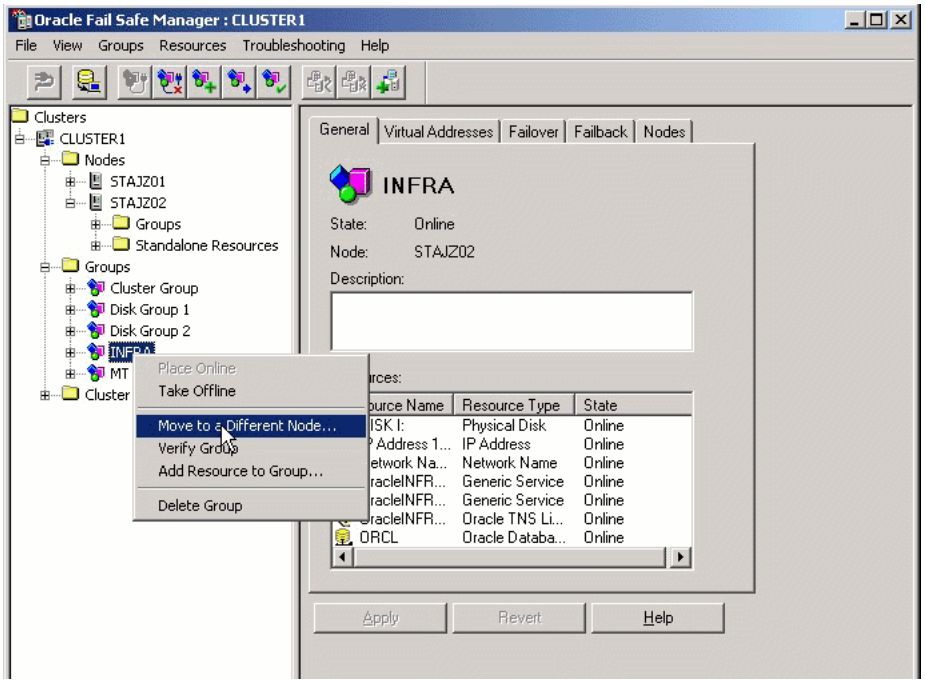


Figure 4–8 Screen 2 Performing Failover for Oracle Identity Management in an Active-Passive Configuration

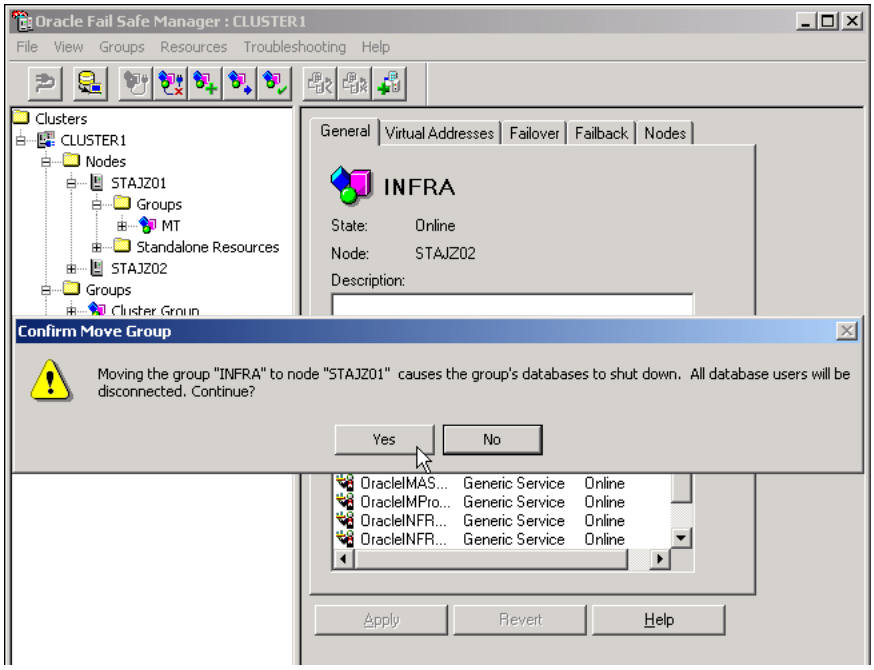
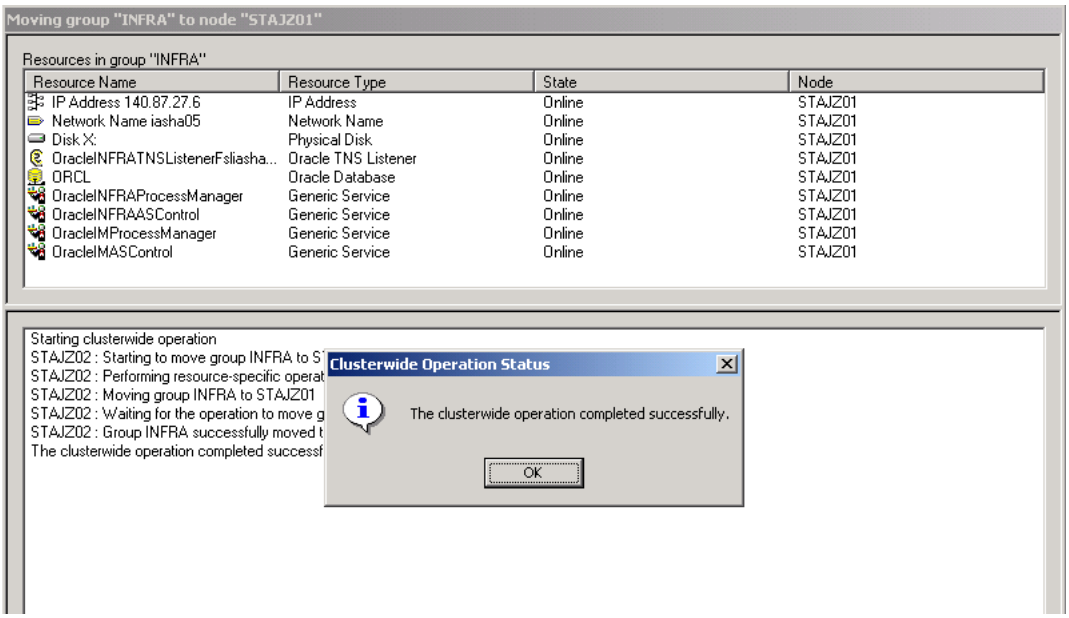


Figure 4–9 Screen 3 Performing Failover for Oracle Identity Management in an Active-Passive Configuration



4.5.2 Failover on Linux Systems

The following shows the steps to failover from the active node to the standby node on Linux systems.

Steps to Perform on the Failed Node

1. Make sure all processes belonging to the OracleAS Cold Failover Cluster (Infrastructure) instance on the failed node are down.
2. Login as root.
3. Use the following command to stop the Oracle Cluster Synchronization Services (CSS) daemon, `ocssd`, if it is running:

   ```
   # /etc/init.d/init.cssd stop
   ```

4. Unmount the file system using the following command:

   ```
   # umount <mount_point>
   ```

   If the file system is busy, check which processes are using the file system with the following command:

   ```
   # fuser -muv <Shared Storage Partition>
   ```

   Stop the processes, if required, using the following command:

   ```
   # fuser -k <Shared Storage Partition>
   ```

5. If the failed node is usable, execute the following command to release the virtual IP address:

   ```
   # ifconfig <interface_name> down
   ```

For example,

```
# ifconfig eth1:1 down
```

Steps to Perform on the New Active Node

1. Login as root.
2. Execute the following command to assign the virtual IP address to this node (the new active node):

   ```
   # ifconfig <interface_name> netmask <subnet_mask> up
   ```

   For example,

   ```
   # ifconfig 144.88.27.125 netmask 255.255.252.0 up
   ```

3. Verify that the virtual IP is up and working using `telnet` from a different host (subnet/domain).
4. Mount the file system using the following command:

   ```
   # mount <Shared Storage Partition> <mount_point>
   ```

   For example:

   ```
   # mount /dev/sdc1 /oracle
   ```

5. If the Oracle Cluster Synchronization Services (CSS) daemon, `ocssd`, is required, run the following command as the user that installed the Oracle home:

   ```
   > /etc/init.d/init.cssd start
   ```

6. Start all OracleAS Infrastructure processes on this new active node with the following commands:
   a. Set the `ORACLE_HOME` environment variable to the OracleAS Infrastructure's Oracle home.
   b. Set the `ORACLE_SID` environment variable to the OracleAS Metadata Repository's system identifier.
   c. Start the OracleAS Metadata Repository database:

      ```
      > ORACLE_HOME/bin/sqlplus /nolog
      SQL> connect SYS as SYSDBA
      SQL> startup
      ```

   d. Start the OracleAS Infrastructure database listener.

      ```
      > ORACLE_HOME/bin/lsnrctl start
      ```

   e. Start OPMN and all OPMN-managed processes using the following command:

      ```
      > ORACLE_HOME/opmn/bin/opmnctl startall
      ```

   f. Start the Application Server Control Console:

      ```
      > ORACLE_HOME/bin/emctl start iasconsole
      ```

4.6 Some Useful Procedures

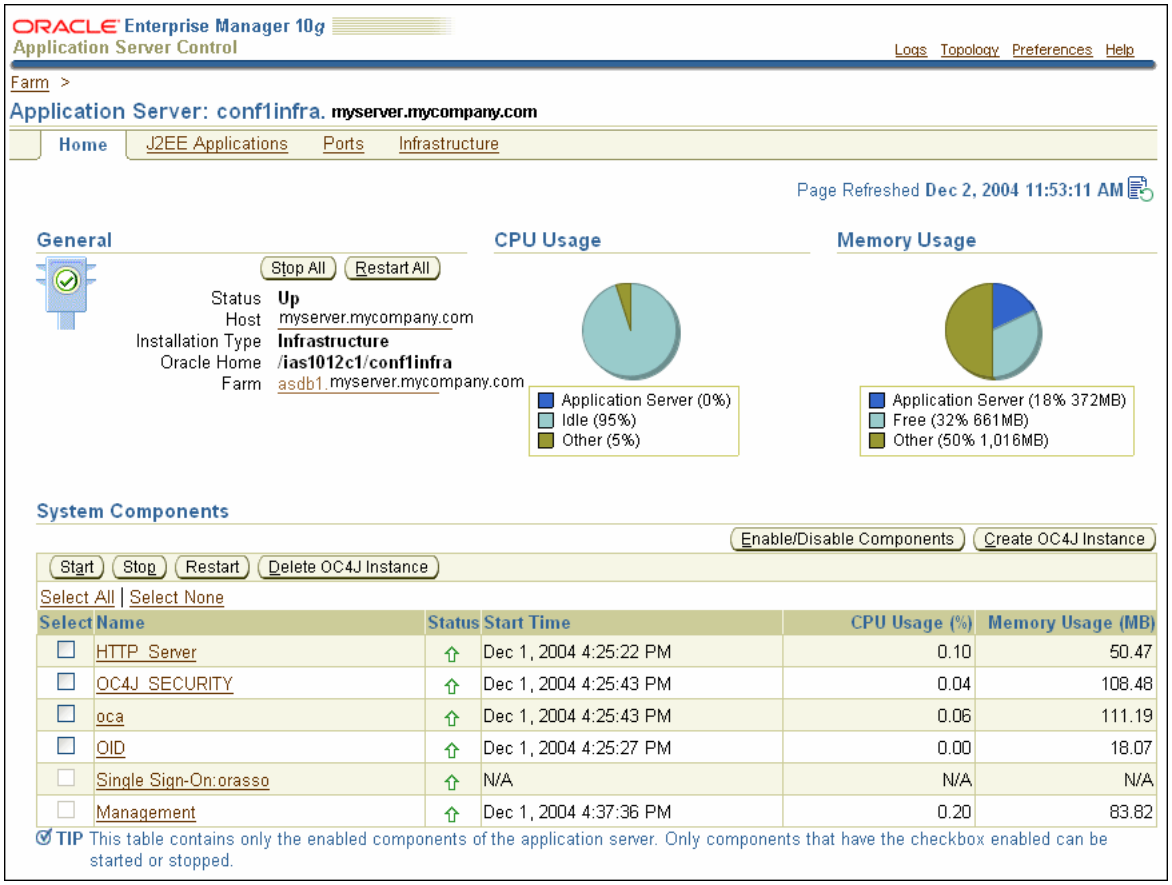
4.6.1 Using Application Server Control Console

You can use the Application Server Control Console to manage OracleAS Cold Failover Cluster topologies. Figure 4–10 shows a sample screen.

To access the Application Server Control Console that is running on hardware cluster nodes, you specify the virtual hostname instead of the physical hostname in the Application Server Control Console URL.

If you are running OracleAS Single Sign-On and Oracle Delegated Administration Services in an active-active mode, you use the physical hostname of either node in the Application Server Control Console URL, for example, `http://sso1.mydomain.com:1156` or `http://sso2.mydomain.com:1156`.

Figure 4–10 Application Server Control Console for OracleAS Cold Failover Cluster (Infrastructure) Topology



4.6.2 Starting the Components

To start Oracle Application Server, you start the components in the following order:

1. Start the OracleAS Metadata Repository database.
2. Start the Oracle Identity Management components.

   If the Oracle Identity Management components are running on different tiers, start them in the following order:

   a. Start Oracle Internet Directory and Oracle Directory Integration Platform.

   b. Start OracleAS Single Sign-On and Oracle Delegated Administration Services.

3. Start Application Server Control Console.

For the active-passive tiers in OracleAS Cold Failover Cluster topologies, make sure that you have done the following steps:

- Enable volume management software and mount the file system on the active node.
- Enable the virtual IP address on the active node.

4.6.2.1 For the OracleAS Cold Failover Cluster (Infrastructure) Topology

Follow these steps to start the OracleAS Infrastructure components in an OracleAS Cold Failover Cluster (Infrastructure) topology:

1. Set the `ORACLE_HOME` environment variable to the OracleAS Infrastructure's Oracle home.
2. Set the `ORACLE_SID` environment variable to the OracleAS Metadata Repository database's system identifier.
3. Set the `PATH` environment variable to include the OracleAS Infrastructure's `ORACLE_HOME/bin` directory.

   On Windows, you can use the following command to set the `PATH`:

   ```
   set PATH=%ORACLE_HOME%\bin;%PATH%
   ```

   Note: Specify the path of the Oracle home as the first entry in the `PATH` environment variable if you have several Oracle homes installed on the computer. Also, ensure that the full paths of the executables you use are specified.

4. Start the OracleAS Metadata Repository database listener.

   ```
   > ORACLE_HOME/bin/lsnrctl start
   ```

5. Start the OracleAS Metadata Repository database:

   On UNIX systems:

   ```
   > $ORACLE_HOME/bin/sqlplus /nolog
   SQL> connect SYS as SYSDBA
   SQL> startup
   ```

   On Windows systems:

   ```
   > %ORACLE_HOME%\bin\sqlplus /nolog
   SQL> connect SYS as SYSDBA
   SQL> startup
   ```

6. Start OPMN and all OPMN-managed processes.

   ```
   > ORACLE_HOME/opmn/bin/opmnctl startall
   ```

7. Start up Application Server Control Console:

   ```
   > ORACLE_HOME/bin/emctl start iasconsole
   ```

4.6.2.2 For the Distributed OracleAS Cold Failover Cluster (Infrastructure) Topology

Follow these steps to start the OracleAS Infrastructure components in a distributed OracleAS Cold Failover Cluster (Infrastructure) topology:

1. On the active node in the OracleAS Metadata Repository and Oracle Internet Directory tier:
   a. Start up the OracleAS Metadata Repository listener and database.
   b. Start up Oracle Internet Directory using OPMN.

      ```
      > ORACLE_HOME/opmn/bin/opmnctl startall
      ```

   c. Start up Application Server Control Console.

      ```
      > ORACLE_HOME/bin/emctl start iasconsole
      ```

2. Start up OracleAS Single Sign-On and Oracle Delegated Administration Services.

   The procedure for starting up these components depends on whether you are running them in an active-active configuration or in an active-passive configuration.

   If you are running them in an active-active configuration, you can follow the procedure described in Section 3.8.2.2, "For the Distributed OracleAS Cluster (Identity Management) Topology", step 3.

   If you are running them in an active-passive configuration, you can start them up using OPMN:

   a. Set the ORACLE_HOME environment variable to the OracleAS Single Sign-On / Oracle Delegated Administration Services home.
   b. Run OPMN to start up the components.

      ```
      > ORACLE_HOME/opmn/bin/opmnctl startall
      ```

   c. Start up Application Server Control Console.

      ```
      > ORACLE_HOME/bin/emctl start iasconsole
      ```

4.6.2.3 For the OracleAS Cold Failover Cluster (Identity Management) Topology

You start the components in the following order:

1. Start up the OracleAS Metadata Repository listener and database.
2. Start up the Oracle Identity Management components using OPMN.

   ```
   > ORACLE_HOME/opmn/bin/opmnctl startall
   ```

3. Start up Application Server Control.

   ```
   > ORACLE_HOME/bin/emctl start iasconsole
   ```

4.6.2.4 For the Distributed OracleAS Cold Failover Cluster (Identity Management) Topology

You start the components in the following order:

1. Start up the OracleAS Metadata Repository listener and database.
2. On the active node for Oracle Internet Directory:

a. Start up Oracle Internet Directory using OPMN.

```
> ORACLE_HOME/opmn/bin/opmnctl startall
```

b. Start up Application Server Control Console.

```
> ORACLE_HOME/bin/emctl start iasconsole
```

3. On each node running OracleAS Single Sign-On and Oracle Delegated Administration Services:

a. Start up OracleAS Single Sign-On, Oracle Delegated Administration Services, and Oracle HTTP Server using OPMN.

```
> ORACLE_HOME/opmn/bin/opmnctl startall
```

b. Start up Application Server Control Console.

```
> ORACLE_HOME/bin/emctl start iasconsole
```

4.6.3 Stopping the Components

To stop Oracle Application Server, you stop the components in the following order:

1. Stop Application Server Control Console.
2. Stop the Oracle Identity Management components.

   If the Oracle Identity Management components are running on different tiers, stop them in the following order:

   a. Stop OracleAS Single Sign-On and Oracle Delegated Administration Services.
   b. Stop Oracle Internet Directory and Oracle Directory Integration Platform.
3. Stop the OracleAS Metadata Repository database.

The next two steps are required only if you are stopping on the current node to fail over to the other node. Otherwise it is not a mandatory part of the stop process.

1. As root, disable volume management software and unmount the file system (if necessary).
2. Disable the virtual IP address from the current node.

4.6.3.1 For the OracleAS Cold Failover Cluster (Infrastructure) Topology

Use the following steps to stop the OracleAS Infrastructure in an OracleAS Cold Failover Cluster (Infrastructure) topology:

1. Set the `ORACLE_HOME` environment variable to the OracleAS Infrastructure's Oracle home.
2. Set the `ORACLE_SID` environment variable to the SID of the OracleAS Metadata Repository database.
3. Stop the Application Server Control Console.

   ```
   > ORACLE_HOME/bin/emctl stop iasconsole
   ```

4. Stop the Oracle Application Server processes using OPMN.

   ```
   > ORACLE_HOME/opmn/bin/opmnctl stopall
   ```

5. Stop the OracleAS Metadata Repository database:

On UNIX systems:

```
> ORACLE_HOME/bin/sqlplus /nolog
SQL> connect SYS as SYSDBA
SQL> shutdown
```

On Windows systems:

```
> ORACLE_HOME\bin\sqlplus /nolog
SQL> connect SYS as SYSDBA
SQL> shutdown
```

6. Stop the OracleAS Metadata Repository database listener.

```
> ORACLE_HOME/bin/lsnrctl stop
```

4.6.3.2 For the Distributed OracleAS Cold Failover Cluster (Infrastructure) Topology

To stop the processes on the different tiers, you stop them in the following order:

1. Stop the processes on each node in the OracleAS Single Sign-On and Oracle Delegated Administration Services tier.
   a. Set the ORACLE_HOME environment variable to the OracleAS Single Sign-On / Oracle Delegated Administration Services home.
   b. Run OPMN to stop the components.

      ```
      > ORACLE_HOME/opmn/bin/opmnctl stopall
      ```

   c. Stop Application Server Control Console.

      ```
      > ORACLE_HOME/bin/emctl stop iasconsole
      ```

2. On the active node in the OracleAS Metadata Repository and Oracle Internet Directory tier:
   a. Set the ORACLE_HOME environment variable to the OracleAS Metadata Repository / Oracle Internet Directory home.
   b. Run OPMN to stop Oracle Internet Directory.

      ```
      > ORACLE_HOME/opmn/bin/opmnctl stopall
      ```

   c. Stop Application Server Control Console.

      ```
      > ORACLE_HOME/bin/emctl stop iasconsole
      ```

   d. Stop the OracleAS Metadata Repository database.

      On UNIX systems:

      ```
      > ORACLE_HOME/bin/sqlplus /nolog
      SQL> connect SYS as SYSDBA
      SQL> shutdown
      ```

      On Windows systems:

      ```
      > ORACLE_HOME\bin\sqlplus /nolog
      SQL> connect SYS as SYSDBA
      SQL> shutdown
      ```

   e. Stop the listener for the OracleAS Metadata Repository database.

```
> ORACLE_HOME/bin/lsnrctl stop
```

4.6.3.3 For the OracleAS Cold Failover Cluster (Identity Management) Topology

To stop the processes, you stop them in the following order:

1. Stop the Oracle Identity Management components using OPMN.
   a. Set the ORACLE_HOME environment variable to the Oracle Identity Management home.
   b. Run OPMN to stop the components.

      ```
      > ORACLE_HOME/opmn/bin/opmnctl stopall
      ```

2. Stop Application Server Control.

   ```
   > ORACLE_HOME/bin/emctl stop iasconsole
   ```

4.6.3.4 For the Distributed OracleAS Cold Failover Cluster (Identity Management) Topology

You stop the processes in the following order:

1. Stop OracleAS Single Sign-On and Oracle Delegated Administration Services:
   a. Set the ORACLE_HOME environment variable to the OracleAS Single Sign-On / Oracle Delegated Administration Services home.
   b. Run OPMN to stop the components.

      ```
      > ORACLE_HOME/opmn/bin/opmnctl stopall
      ```

   c. Stop Application Server Control.

      ```
      > ORACLE_HOME/bin/emctl stop iasconsole
      ```

2. Stop Oracle Internet Directory:
   a. Set the ORACLE_HOME environment variable to the Oracle Internet Directory home.
   b. Run OPMN to stop Oracle Internet Directory.

      ```
      > ORACLE_HOME/opmn/bin/opmnctl stopall
      ```

   c. Stop Application Server Control Console.

      ```
      > ORACLE_HOME/bin/emctl stop iasconsole
      ```

4.6.4 Configuring Components in OracleAS Cold Failover Cluster Topologies

For components that run on active-passive tiers, you can use the standard administration techniques described in the *Oracle Application Server Administrator's Guide*. This is because you are running only one Oracle Application Server instance at any time (only the active node runs the Oracle Application Server instance), and you have only one Oracle home to manage.

Remember that to access the Application Server Control Console, you use the virtual hostname in the Application Server Control Console URL. See Section 4.6.1, "Using Application Server Control Console" for details.

4.6.5 Configuring Virtual IPs

The *Oracle Application Server Installation Guide* for your platform covers the instructions for configuring the virtual IPs for OracleAS Cold Failover Cluster topologies:

- If you are running on UNIX platforms, see section 11.2.2, "Map the Virtual Hostname and Virtual IP Address" in the *Oracle Application Server Installation Guide* for your platform.
- If you are running on Microsoft Windows, see section 11.2.2, "Get a Virtual Address for the Cluster" in the *Oracle Application Server Installation Guide* for Windows.

5

High Availability for Oracle Access Manager

This chapter describes high availability topologies for Oracle Access Manager. This chapter contains the following sections:

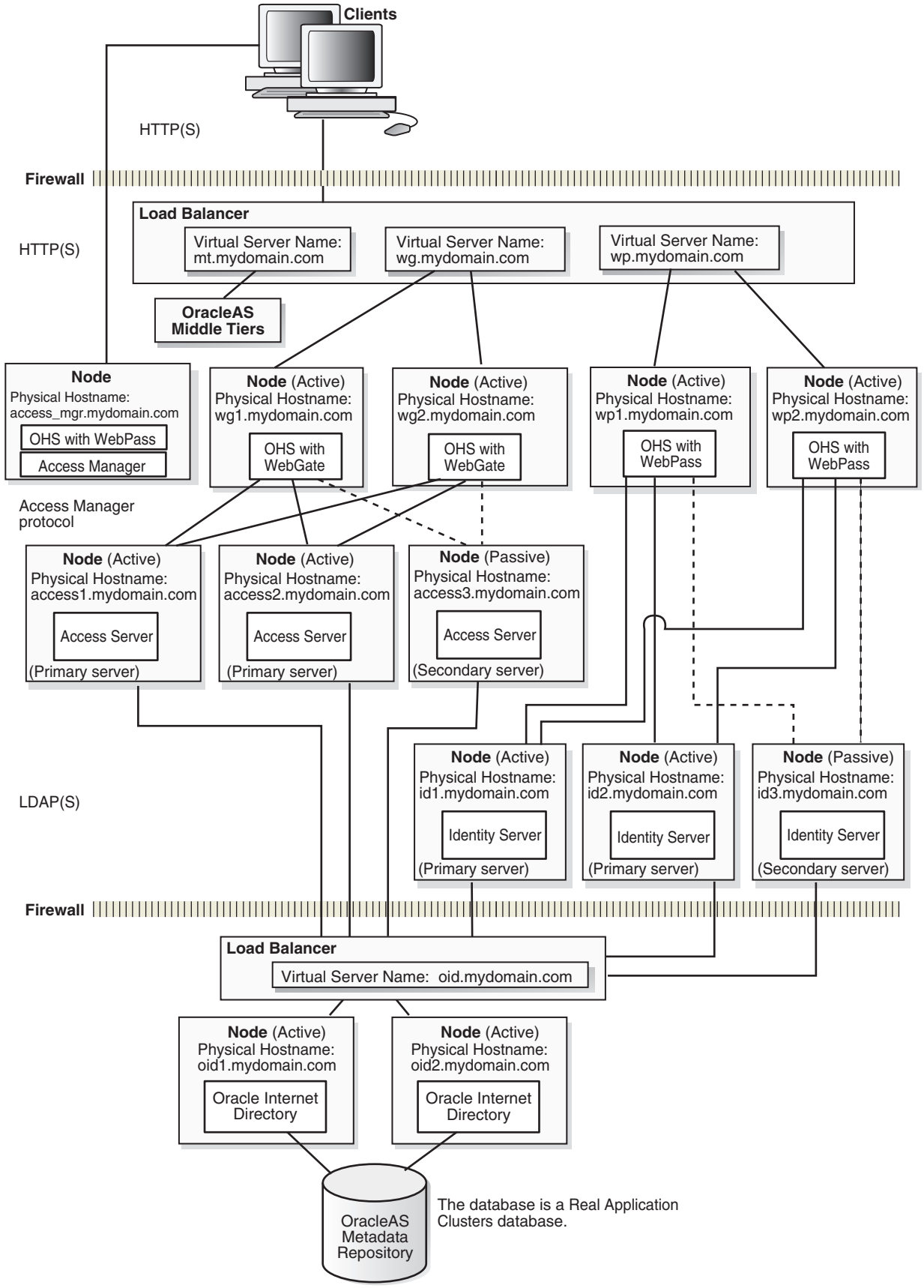
5.1 Overview of High Availability Topologies for Oracle Access Manager

To run Oracle Access Manager in a high availability manner, you can run Oracle Access Manager in an active-active topology, as shown in Figure 5–1. This topology has the following features:

- You need a load balancer to direct traffic to the Oracle HTTP Servers installed with WebGate and WebPass. Clients use the virtual hostname configured on the load balancer to access WebGate and WebPass.
- WebGate and WebPass connect to primary and secondary Access Servers and Identity Servers. You cannot use a load balancer between WebGate/WebPass and Access or Identity Servers because communication between these components uses the proprietary Oracle Access and Oracle Identity protocols.
- You need to set up multiple Access Servers and Identity Servers and configure them as primary and secondary so that WebGate and WebPass can redirect requests to other primary servers or to secondary servers if the primary servers are unavailable.
- When using Oracle Internet Directory with Oracle Access Manager in a high availability environment, you can set up Oracle Internet Directory in different ways for high availability. You can set up Oracle Internet Directory in an active-active or active-passive topology, or you can also set it up with multimaster replication. The replica can be considered a secondary server.

Note that you can configure Oracle Access Manager in an active-active topology only. Active-passive topology for Oracle Access Manager is not supported. Components used by Oracle Access Manager, such as Oracle Internet Directory, may be configured in an active-passive topology, if supported by that component.

Figure 5–1 Oracle Access Manager in Active-Active Topology



5.2 Installing Oracle Access Manager in a High Availability Topology

To install and configure Oracle Access Manager in a high availability topology, see chapter 7, "Installing and Configuring myJ2EE with Oracle Access Manager", in the *Oracle Application Server Enterprise Deployment Guide*.

5.3 Managing Oracle Access Manager in a High Availability Topology

To manage Oracle Access Manager in a high availability topology, you use the same tools as in a non-high availability topology. For example, you use the Identity System Console to configure the Identity System.

The URLs for accessing the tools remain the same. For example:

- To access the Access System Console, use the following URL:
  `http://hostname:port/access/oblix`.

  `hostname` refers to the node running WebGate, and *`port`* refers to the Oracle HTTP Server port.

- To access the Identity System Console, use the following URL:
  `http://hostname:port/identity/oblix`.

  `hostname` refers to the node running WebPass, and *`port`* refers to the Oracle HTTP Server port.

For configuring Oracle Access Manager in a high availability topology, you should be familiar with the following features and procedures:

- Section 5.3.1, "Adding Identity Servers and WebPass Instances"
- Section 5.3.2, "Adding AccessGates and Access Servers"
- Section 5.3.3, "Clustering Access Servers"
- Section 5.3.4, "Associating AccessGate with an Access Server Cluster"
- Section 5.3.5, "Configuring Load Balancing and Failover for Oracle Access Manager Components"
- Section 5.3.6, "Managing Oracle Access Manager Processes"

Details on configuring Oracle Access Manager are provided in the Oracle Access Manager guides.

5.3.1 Adding Identity Servers and WebPass Instances

You may need to add Identity Servers and/or WebPass instances to your system. A WebPass can be associated with one or more Identity Server, and one Identity Server can receive requests from one or more WebPass instances.

For details on adding Identity Servers, see section 7.4, "Managing Identity Servers", in the *Oracle Access Manager Identity and Common Administration Guide*.

For details on adding WebPass instances, see section 7.7, "Configuring WebPass", in the *Oracle Access Manager Identity and Common Administration Guide*.

5.3.2 Adding AccessGates and Access Servers

To add AccessGates to your system, see section 2.4.3, "Adding an AccessGate", in the *Oracle Access Manager Access System Administration Guide*.

To add Access Servers to your system, see section 2.3.2, "Adding an Access Server Instance", in the *Oracle Access Manager Access System Administration Guide*.

5.3.3 Clustering Access Servers

You should cluster your Access Servers for the following reasons:

- You can associate an AccessGate with one or more Access Server clusters. This enables the AccessGate to fail over to another Access Server in the cluster if the first Access Server is not available.
- Oracle Access Manager automatically configures failover and load balancing for all the Access Servers in a cluster.
- You can configure a cluster to be a primary cluster or a backup cluster. AccessGate creates connections to the Access Servers in the backup cluster if it is unable to create connections to the Access Servers in the primary cluster.

Note that all Access Servers in a cluster and all AccessGates associated with the cluster must have the same transport security mode and Policy API Support mode.

See section 2.3.5, "Clustering Access Servers", in the *Oracle Access Manager Access System Administration Guide* for the steps on how to cluster the Access Servers.

5.3.4 Associating AccessGate with an Access Server Cluster

To associate an AccessGate with an Access Server cluster, see section 2.6, "Associating AccessGates with Access Servers", in the *Oracle Access Manager Access System Administration Guide*.

5.3.5 Configuring Load Balancing and Failover for Oracle Access Manager Components

Oracle Access Manager can perform both load balancing and failover between these components:

- From WebPass to Identity Servers
- From WebGate to Access Servers
- From Identity Server to directory servers
- From Access Server to directory servers

In addition, you can configure failover for Policy Manager to fail over to a secondary directory server. Load balancing for Policy Manager is not supported.

For load balancing and failover, you designate the Identity Servers, Access Servers, and directory servers as primary or secondary. Oracle Access Manager creates connections to secondary servers only if connections to the primary servers become unavailable.

You can also cluster the Identity Servers and Access Servers, if you want. Clustering is recommended for active-active topologies.

Using Hardware Load Balancer vs. the Load Balancing Feature in Oracle Access Manager

Generally, if you already have a hardware load balancer in front of your Oracle Internet Directory for reasons not related to Oracle Access Manager, the best option is to use the hardware load balancer as the only load balancing mechanism. This option

is probably more efficient in that the hardware load balancer offloads the load balancing tasks from Oracle Access Manager and is easier to maintain.

Some examples where you might need a hardware load balancer in front of Oracle Internet Directory:

- You are already running a previous release of Oracle Internet Directory with a hardware load balancer, and you have users accessing this Oracle Internet Directory only through this hardware load balancer.
- OracleAS Portal does not load balance requests to Oracle Internet Directory automatically. If you are using OracleAS Portal with Oracle Internet Directory, then you are going to require a hardware load balancer.

In these examples (where a hardware load balancer exists for other reasons), then you should use the hardware load balancer to load balance requests to Oracle Internet Directory.

However, if you do not have a hardware load balancer, and you do not have other components that require a hardware load balancer to access a redundant Oracle Internet Directory, you can configure the load balancing feature in Oracle Access Manager.

Load balancing and failover for Oracle Access Manager are described in the "Failover and Load Balancing" chapter in the *Oracle Access Manager Deployment Guide*.

5.3.6 Managing Oracle Access Manager Processes

WebPass and WebGate instances, because they run within Oracle HTTP Server, are managed by OPMN. If an Oracle HTTP Server process dies or becomes unavailable, OPMN tries to restart it.

Identity Server and Access Server are not monitored by OPMN. You will have to manage these servers yourself.

Oracle Internet Directory is managed by OPMN.

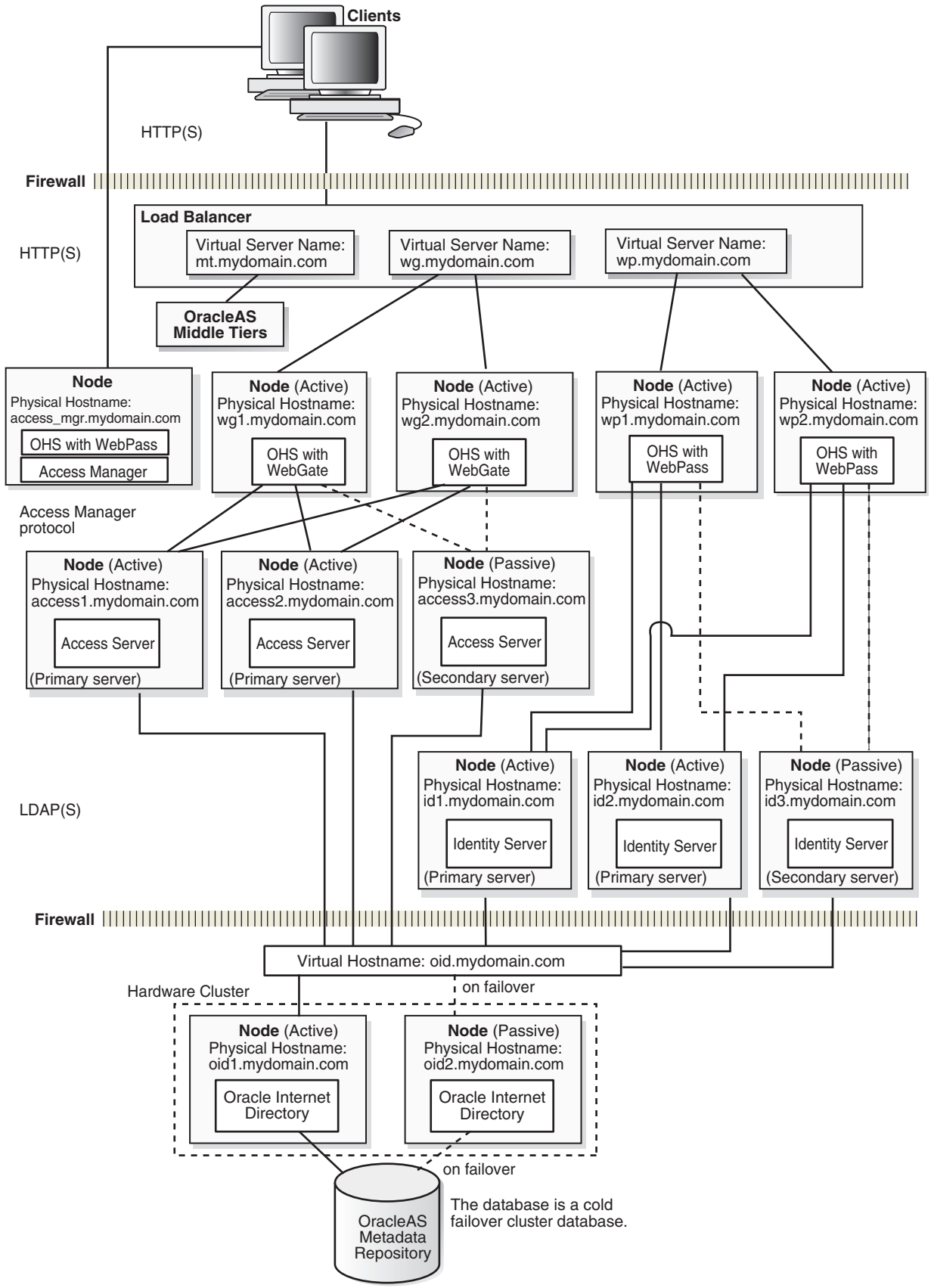
5.4 Configuring Oracle Internet Directory in an Active-Passive Topology for Oracle Access Manager

While Oracle Access Manager must be configured in an active-active topology, components that it uses can be configured in different topologies. For example, Oracle Internet Directory can be configured in an active-active (shown in Figure 5–1) or active-passive (shown in Figure 5–2) topology. In the two figures, the Oracle Access Manager topology is unchanged, the only difference is in the Oracle Internet Directory configuration.

When Oracle Internet Directory is running in an active-passive topology, it also uses a cold failover cluster database, as shown in Figure 5–2.

To install the Oracle Internet Directory in an active-passive topology, see the "Installing in High Availability Environments: OracleAS Cold Failover Cluster" chapter in the *Oracle Application Server Installation Guide* for your platform.

Figure 5–2 Oracle Access Manager with Oracle Internet Directory in Active-Passive Topology



6

High Availability for Oracle Identity Federation

This chapter describes how to run Oracle Identity Federation in an OracleAS Cold Failover Cluster (or active-passive) topology. Running Oracle Identity Federation in an active-active configuration is not supported.

This chapter contains the following sections:

6.1 OracleAS Cold Failover Cluster Topology for Oracle Identity Federation

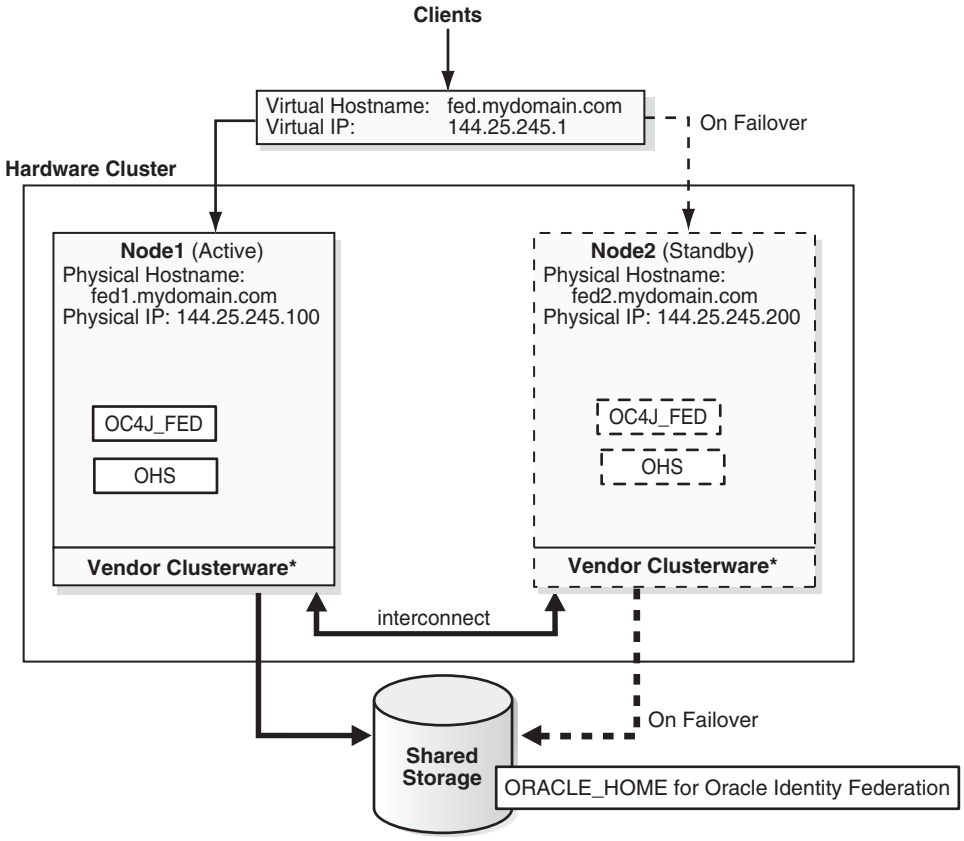
Figure 6–1 shows a diagram of Oracle Identity Federation in an OracleAS Cold Failover Cluster topology. Oracle Identity Federation runs within the OC4J_FED instance.

The OracleAS Cold Failover Cluster topology for Oracle Identity Federation is similar to other OracleAS Cold Failover Cluster topologies for Oracle Application Server:

- You have two nodes in a hardware cluster, and these nodes need to be running vendor clusterware. If you are running on Windows, these nodes need to be running Microsoft Cluster Server and Oracle Fail Safe.
- The two nodes have access to a shared storage. You install the Oracle home for Oracle Identity Federation on the shared storage.
- Only one node is active at any time. The other node is passive. If the active node fails, a failover event occurs: the passive node mounts the shared storage and runs the processes that were running on the active node.
- The two nodes are associated with a virtual hostname and virtual IP address. Clients use the virtual hostname to access the components running on the hardware cluster.

  Using the virtual hostname enables the client to be unaware of which node is actually processing the request.

Figure 6–1 Oracle Identity Federation in OracleAS Cold Failover Cluster Topology



* On Windows, you need Microsoft Cluster Server and Oracle Fail Safe.

6.1.1 Installing Oracle Identity Federation in an OracleAS Cold Failover Cluster Topology on Linux

To install and configure Oracle Identity Federation in an OracleAS Cold Failover Cluster topology on Linux, perform the following steps:

Note: These steps are similar to those for installing an OracleAS Cold Failover Cluster (Identity Management) topology. The only difference is that you install Oracle Identity Federation in step 2 below instead of Oracle Identity Management.

The procedure below provides only high-level steps. For detailed steps, see the referenced section in the *Oracle Application Server Installation Guide for Linux*.

---

1. Perform pre-installation tasks:
   - Map the virtual hostname and IP address. [See *Oracle Application Server Installation Guide for Linux*, section 9.2.1, "Map the Virtual Hostname and Virtual IP Address".]
   - Set up a file system on the shared storage that can be mounted from either node. [See *Oracle Application Server Installation Guide for Linux*, section 9.2.2, "Set Up a File System That Can Be Mounted from Both Nodes".]

- If you are planning to use the Automatic Storage Management feature in the Oracle database, see *Oracle Application Server Installation Guide for Linux*, section 9.2.3, "Review recommendations for Automatic Storage Management".
- Check that the clusterware is running. [See *Oracle Application Server Installation Guide for Linux*, section 9.2.4, "Check That Clusterware Is Running".]

2. Install Oracle Identity Federation on the shared storage. See the "Installing Oracle Identity Federation" chapter in the *Oracle Identity Federation Administrator's Guide*.

Installation Notes: To run Oracle Identity Federation in an OracleAS Cold Failover Cluster topology, you need to select the Advanced option in the installer. This enables you to:

- Configure Oracle Identity Federation to store its data in an LDAP server and Oracle database. See Section 6.1.3, "Configuring Data Store for Oracle Identity Federation" for details.
- Configure a virtual hostname for Oracle Identity Federation. See Section 6.1.4, "Configuring Virtual Addressing" for details.

6.1.2 Installing Oracle Identity Federation in an OracleAS Cold Failover Cluster Topology on Windows

On Windows, the installation procedure involves more steps because you have to install Oracle Fail Safe on both nodes in the hardware cluster, and you also have to make sure that the Windows registry on both nodes are set up correctly for Oracle Fail Safe and Oracle Application Server.

On Windows, you need to install and run Oracle Fail Safe on the nodes in the hardware cluster. These nodes must already be running Microsoft Cluster Server. Oracle Fail Safe and Microsoft Cluster Server act as the clusterware for the hardware cluster. They monitor the hardware as well as the software running on the nodes.

To install and configure Oracle Identity Federation in an OracleAS Cold Failover Cluster topology on Windows, perform the following steps:

Note: These steps are similar to those for installing an OracleAS Cold Failover Cluster (Identity Management) topology. The only difference is that you install Oracle Identity Federation in step 5 below instead of Oracle Identity Management.

The procedure below provides only high-level steps. For detailed steps, see the referenced section in the *Oracle Application Server Installation Guide for Windows*.

---

1. Perform pre-installation tasks:
   - Ensure that the Event Log Service is running. [See *Oracle Application Server Installation Guide for Windows*, section 9.2.1, "Ensure that the Event Log Service Is Running".]
   - Determine the virtual address for the hardware cluster. [See *Oracle Application Server Installation Guide for Windows*, section 9.2.2, "Get a Virtual Address for the Cluster".]

- Verify that Microsoft Cluster Server (MSCS) is installed on both nodes in the hardware cluster. [See *Oracle Application Server Installation Guide for Windows*, section 9.2.3, "Verify that Microsoft Cluster Server (MSCS) Is Installed on Both Nodes".]
- Determine the name of the cluster. [See *Oracle Application Server Installation Guide for Windows*, section 9.2.4, "Determine the Name of the Cluster".]
- Determine the domain user to administer Oracle Fail Safe. [See *Oracle Application Server Installation Guide for Windows*, section 9.2.5, "Determine a Domain User to Administer Oracle Fail Safe".]

2. Install Oracle Fail Safe on the local storage of each node in the hardware cluster. [See *Oracle Application Server Installation Guide for Windows*, section 9.2.6, "Install Oracle Fail Safe on the Local Storage of Each Node".]
3. Create a group in Oracle Fail Safe, and add these resources to the group: [See *Oracle Application Server Installation Guide for Windows*, section 9.2.7, "Create a Group in Oracle Fail Safe".]
   - Virtual IP address
   - Virtual hostname
   - Shared disk. You will add this using the Cluster Administrator tool in the next step.
4. Use the Cluster Administrator tool to add the shared disk to the group you created in Oracle Fail Safe. [See *Oracle Application Server Installation Guide for Windows*, section 9.2.7, "Create a Group in Oracle Fail Safe".]
5. From node 1, install Oracle Identity Federation on the shared disk in the hardware cluster. See the "Installing Oracle Identity Federation" chapter in the *Oracle Identity Federation Administrator's Guide*.

Installation Notes: To run Oracle Identity Federation in an OracleAS Cold Failover Cluster topology, you need to select the Advanced option in the installer. This enables you to:

- Configure Oracle Identity Federation to store its data in an LDAP server and Oracle database. See Section 6.1.3, "Configuring Data Store for Oracle Identity Federation" for details.
- Configure a virtual hostname for Oracle Identity Federation. See Section 6.1.4, "Configuring Virtual Addressing" for details.

---

6. On node 1, stop all processes and services running out of the Oracle Identity Federation Oracle home. Also, configure the services' startup type to Manual. [See *Oracle Application Server Installation Guide for Windows*, step 4, "Stop the Oracle Application Server Services on Node 1, and Set Their Startup Type to Manual", in section 9.5.2, "OracleAS Cold Failover Cluster (Identity Management): Details of Installation Steps".]
7. Configure node 2 in the hardware cluster. This is to ensure that it is configured similarly for Oracle Identity Federation. [See *Oracle Application Server Installation Guide for Windows*, step 5, "Configure Node 2", in section 9.5.2, "OracleAS Cold Failover Cluster (Identity Management): Details of Installation Steps".]

If the nodes in the hardware cluster are symmetrical, you can run some scripts to configure node 2 so that the registry settings, service settings, and Oracle inventory settings are identical on both nodes.

If the nodes in the hardware cluster are asymmetrical, you need to install Oracle Identity Federation again, but this time from node 2. The installer then configures the registry on node 2. Before you perform the installation from node 2, you need to delete the first installation from the shared disk because you will be installing it again from the other node.

8. Restart node 2. [See *Oracle Application Server Installation Guide for Windows*, step 6, "Restart Node 2", in section 9.5.2, "OracleAS Cold Failover Cluster (Identity Management): Details of Installation Steps".]

9. Move the group that you created in Oracle Fail Safe to node 2. [See *Oracle Application Server Installation Guide for Windows*, step 7, "Move the Group to Node 2", in section 9.5.2, "OracleAS Cold Failover Cluster (Identity Management): Details of Installation Steps".]

10. Start up the Oracle Identity Federation processes and services on node 2.

    You can run `opmnctl` to do this:

    ```
    > opmnctl startall
    ```

    This step is equivalent to *Oracle Application Server Installation Guide for Windows*: step 8, "Start up OracleAS Infrastructure Services on Node 2", in section 9.5.2, "OracleAS Cold Failover Cluster (Identity Management): Details of Installation Steps".

11. Verify the installation by accessing the Application Server Control Console and the Oracle Identity Federation Administration Control pages in a browser. In the URL for these pages, you use the virtual hostname, not the physical hostname. For example:

    - The URL for Application Server Control Console might look like http://fed.mydomain.com:18103, assuming "fed.mydomain.com" is the virtual hostname, and 18103 is the port for Application Server Control Console.
    - The URL for Oracle Identity Federation Administration Console might look like `http://fed.mydomain.com:7779/fedadmin`, assuming "fed.mydomain.com" is the virtual hostname, and 7779 is the port for Oracle HTTP Server.

    This step is equivalent to *Oracle Application Server Installation Guide for Windows*, step 9, "Verify Installation", in section 9.5.2, "OracleAS Cold Failover Cluster (Identity Management): Details of Installation Steps".

12. Add OPMN and Application Server Control Console to the list of processes that are to be monitored by Oracle Fail Safe, and also make the shared disk a dependency for OPMN. See the following sections in the *Oracle Application Server Installation Guide for Windows* for step details.

    *S*ection 9.10.4, "Make OPMN Highly Available"

    Section 9.10.5, "Add the Shared Disk as a Dependency for OPMN"

    Section 9.10.6, "Make Application Server Control Console Highly Available"

6.1.3 Configuring Data Store for Oracle Identity Federation

To run Oracle Identity Federation in an OracleAS Cold Failover Cluster topology:

- You need to configure Oracle Identity Federation to store federation data in an LDAP server (such as Oracle Internet Directory) or in a database (such as Oracle Database), instead of storing the data in memory.
- You also need to configure Oracle Identity Federation to store transient data in a database, instead of storing the data in memory.

If you choose to store the federation and transient data in memory, the data would be lost if a node in the OracleAS Cold Failover Cluster topology goes down.

To configure Oracle Identity Federation to store the data in an LDAP server and Oracle database, select the Advanced option during installation, and then select the "Federation Data in LDAP Server" and "Federation Transient Data in Database" options. The installer displays screens in which you enter connect information for an LDAP server and Oracle database.

To ensure that the entire system is highly available, you should ensure that the backend servers (that is, the LDAP server and the database) used by Oracle Identity Federation are also highly available. For example, the database can be an Oracle Real Application Clusters (Oracle RAC) database.

6.1.4 Configuring Virtual Addressing

To run Oracle Identity Federation in an OracleAS Cold Failover Cluster topology, you need to configure Oracle Identity Federation with a virtual hostname. The virtual hostname (as opposed to a physical hostname) enables Oracle Identity Federation to run on either node in the hardware cluster.

You can configure the virtual hostname during installation by selecting the Advanced option, and selecting the Virtual Addressing Option. You then enter the virtual hostname in the Specify Virtual Hostname screen.

6.1.5 Monitoring Processes and Failing Over

Clusterware is needed on the nodes to monitor the health of the nodes. Clusterware is usually provided by the hardware vendor. If the active node fails, clusterware helps in failing over resources (such as the shared storage and the virtual hostname and IP address) to the passive node.

On Windows, you need Oracle Fail Safe and Microsoft Cluster Server as the clusterware.

Process Monitoring in Oracle Application Server

In Oracle Application Server, Oracle Process Manager and Notification Server (OPMN) monitors the OC4J_FED instance and Oracle HTTP Server. The OC4J_FED instance is the OC4J instance that runs Oracle Identity Federation.

If the OC4J_FED instance or Oracle HTTP Server fails, OPMN tries to restart it. If the restart fails, then the clusterware (Oracle Fail Safe, if on Windows) fails over all the processes to the passive node in the hardware cluster. Clients may experience a brief disruption of service, but after the failover is complete, clients should be able to access Oracle Identity Federation as usual.

6.2 Fast Connection Failover for Oracle Identity Federation

If your Oracle Identity Federation uses an Oracle 10*g* Oracle RAC database, you can configure Oracle Identity Federation to use the fast connection failover feature. Fast connection failover provides rapid detection and cleanup of invalid cached connections, and load balancing of available connections. For more information about fast connection failover, see the "Fast Connection Failover" chapter in the *Oracle Database JDBC Developer's Guide and Reference*.

To use fast connection failover, you also need to enable the implicit connection cache. Implicit connection caching is described in the "Implicit Connection Caching" chapter in the *Oracle Database JDBC Developer's Guide and Reference*.

To enable fast connection failover in Oracle Identity Federation:

1. Insert the following line in the `<data-source>` element of the `ORACLE_HOME/j2ee/OC4J_FED/config/data-sources.xml` file for the OC4J_FED instance (which is the OC4J instance that runs Oracle Identity Federation):

   ```
   <property name='fastConnectionFailoverEnabled' value='true'/>
   ```

2. Update the `ORACLE_HOME/opmn/conf/ons.conf` file in the Oracle Identity Federation Oracle home as follows:

   a. On the `localport` line, set the local port for the ONS daemon. The local port is used by local clients to communicate with the ONS daemon:

      ```
      localport=ONS_LOCAL_PORT
      ```

      You can determine the port number from the `ORACLE_HOME/opmn/conf/opmn.xml` file.

   b. On the `remoteport` line, set the remote port, which is the port used by other ONS daemons to communicate with this ONS daemon:

      ```
      remoteport=ONS_REMOTE_PORT
      ```

      You can determine the port number from the `ORACLE_HOME/opmn/conf/opmn.xml` file.

   c. On the `nodes` line, set the list of nodes that are running the other ONS daemons with which this ONS daemon needs to communicate.

      The format is a comma-delimited list of *RAC_NODE*:*ONS_REMOTE_PORT*. For example:

      ```
      nodes=RAC_NODE1:RAC_NODE1_ONS_REMOTE_PORT,RAC_NODE2:RAC_NODE2_ONS_REMOTE_PORT
      ```

      You can determine the remote port used by an ONS daemon by looking in the `RAC_ORACLE_HOME/opmn/conf/ons.config` file.

3. Restart Oracle Identity Federation.

   ```
   ORACLE_HOME/opmn/bin/opmnctl stopall
   ORACLE_HOME/opmn/bin/opmnctl startall
   ```

4. Update Oracle Identity Federation ONS details in the Oracle Cluster Registry (OCR) using the `racgons` command.

   a. In the Oracle home for the Oracle RAC database, navigate to the `bin` directory.

      ```
      cd RAC_ORACLE_HOME/bin
      ```

b. Run the following command:

```
racgons add_config FedServer_Node:FedServer_Node_ONS_Remote_Port
```

You can determine the ONS remote port on the Oracle Identity Federation node by looking in the `ORACLE_HOME/opmn/conf/opmn.xml` file, where Oracle home is the installation directory for Oracle Identity Federation.

5. Test ONS by running the following command in the Oracle Identity Federation Oracle Home:

```
ORACLE_HOME/opmn/bin/opmnctl debug
```

In the output, you should see the IP addresses of the Oracle RAC database nodes under "ONS Server Connections".

7

High Availability for OracleAS Metadata Repository

This chapter describes high availability configurations for the OracleAS Metadata Repository. To make OracleAS Metadata Repository highly available, you need to make the database highly available. Common configurations for a highly available database include:

- Section 7.1, "Cold Failover Cluster Databases"
- Section 7.2, "Oracle Real Application Clusters Databases"
- Section 7.3, "Other High Availability Solutions for the OracleAS Metadata Repository Database"

The last section, Section 7.4, "Checking the Status of OracleAS Metadata Repository", describes how to check the status of the database and the listener.

7.1 Cold Failover Cluster Databases

In a cold failover cluster database, you install the database on storage shared by two nodes. These nodes are in a hardware cluster. One of these nodes (the active node) runs the database processes. If the active node fails for any reason, a failover event occurs and the other node (the passive node) takes over and runs the database processes.

In a cold failover cluster configuration, you set up a virtual hostname and virtual IP address, and associate them with the active node. Clients access the active node using the virtual hostname and virtual IP address. Using a virtual hostname or virtual IP address shields the clients from needing to know which node is servicing their requests. During a failover, the passive node becomes the active node, and the virtual hostname and virtual IP address are now associated with the new active node.

Figure 7–1 shows a diagram of a cold failover cluster database on UNIX; Figure 7–2 shows it on Windows. Note the following differences:

- On Windows, you need to run Oracle Fail Safe and Microsoft Cluster Server. On UNIX, you run the vendor clusterware.
- On Windows, you install the database ORACLE_HOME on the local storage of each node. On UNIX, you install it on the shared storage. For both UNIX and Windows, the data files for the database are located on the shared storage.

Figure 7–1 Cold Failover Cluster Database on UNIX

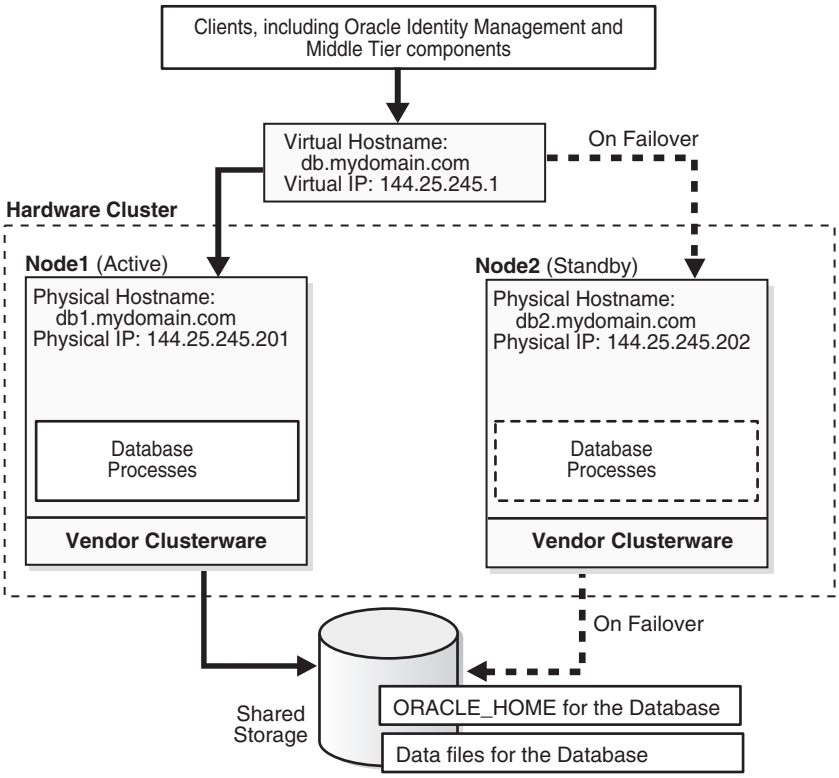
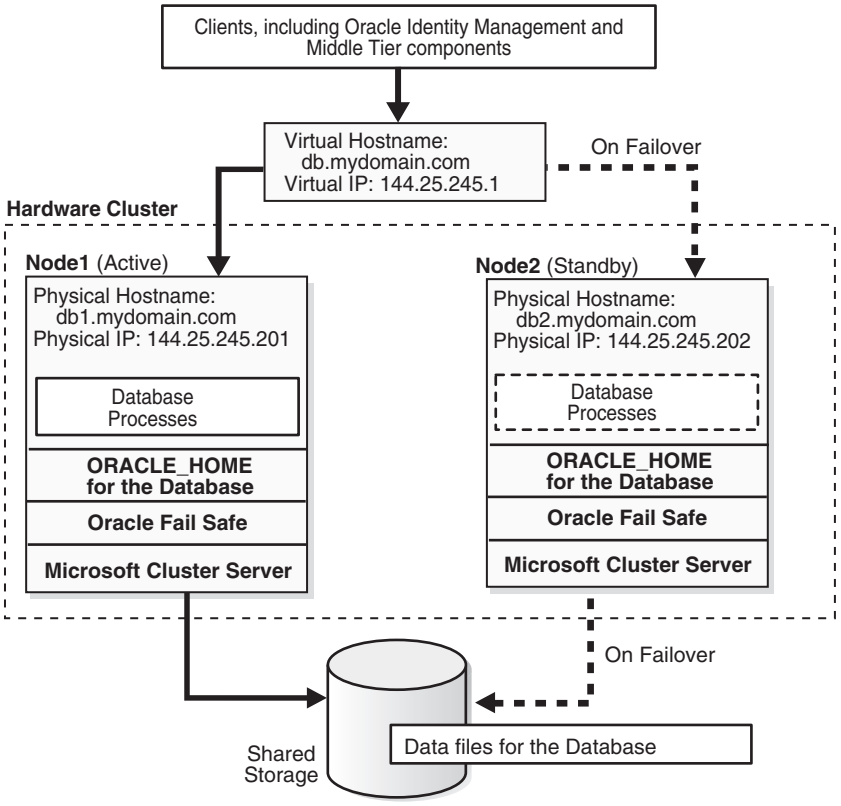


Figure 7–2 Cold Failover Cluster Database on Windows



7.1.1 Installing a Cold Failover Cluster Database

You can use the Oracle Application Server installer to install a cold failover cluster database that is already populated with the OracleAS Metadata Repository. See the *Oracle Application Server Installation Guide* for details.

If you already have a cold failover cluster database, you can load the OracleAS Metadata Repository into the existing database using the OracleAS Metadata Repository Creation Assistant. See the *Oracle Application Server Metadata Repository Creation Assistant User's Guide* for details.

7.1.2 Running a Cold Failover Cluster Database

Running and managing a cold failover cluster database is similar to running single-instance database. The only difference is using the virtual hostname instead of the physical hostname.

7.1.3 Running Database Console against a Cold Failover Cluster Database

Before you can start, stop or check the status of Database Console against a cold failover cluster database, you need to set the ORACLE_HOSTNAME environment variable to the virtual hostname. This is required on UNIX platforms. For example, in Figure 7–1, the virtual hostname is db.mydomain.com. You would set ORACLE_HOSTNAME as follows:

C shell:

```
$ setenv ORACLE_HOSTNAME db.mydomain.com
```

Bourne or Korn shell:

```
% ORACLE_HOSTNAME=db.mydomain.com
% export ORACLE_HOSTNAME
```

After setting the variable, you can then run the "`emctl` *`action`* `dbconsole`" commands, where *`action`* is `start`, `stop`, or `status` (for example, `emctl start dbconsole`).

On Windows, ORACLE_HOSTNAME is set in the Windows registry. You do not need to set it as an environment variable.

7.1.4 Backing Up a Cold Failover Cluster Database

Backup and recovery procedures are covered in detail in the *Oracle Application Server Administrator's Guide*. This section describes backup and recovery details that are specific to cold failover cluster databases.

Backup Considerations

- Place archive logs for the OracleAS Metadata Repository on the shared disk. This ensures that when failing over from one cluster node to another in the case of media recovery, the archive logs are also failed over and available.

  You can generate archive logs to a local file system. However, make this destination accessible to both nodes so that no matter which node is active, the database instance writes archive logs to the same location. Otherwise, the backup operation will not be able to see all the archive log files.

- Plan your capacity requirements carefully to ensure that you have adequate space to store the desired number of archive logs.

Recovery Considerations

There are no special considerations for recovering a cold failover cluster database. If you store archive logs on a local file system, in the case of media recovery, you must make all archive logs be available to the Oracle Application Server instance performing the recovery. You can perform the recovery from either node of the cluster.

If You Are Running on Microsoft Windows

One of the steps that you have to do to prepare your OracleAS Metadata Repository for backup using the OracleAS Backup and Recovery Tool is to run the "`alter database archivelog`" command to enable ARCHIVELOG mode. See the "Enabling ARCHIVELOG Mode" section in the *Oracle Application Server Administrator's Guide*.

However, if Oracle Fail Safe has database polling enabled, the following error message will appear when you run the command:

```
ORA-01126: database must be mounted EXCLUSIVE and not open for this operation
```

To avoid this error message, you need to disable database polling in Oracle Fail Safe. To do this:

1. Start Oracle Fail Safe Manager.
2. Expand the following: Clusters > *cluster_name* > Cluster Resources, and select *db_instance_name*.

   cluster_name is the name of the cold failover cluster, and *db_instance_name* is the name of the database instance.
3. Select the Database tab.
4. Disable Database Polling.

After completing the backup or restore operation, you can re-enable database polling.

You need to disable database polling because the OracleAS Backup and Recovery Tool shuts down the database. On Windows, Oracle Fail Safe performs database polling and restarts the database if it is down. This means that every time before you perform "backup_cold" or "restore_repos" with the OracleAS Backup and Recovery Tool on the active node, you must disable database polling in the Oracle Fail Safe Manager and re-enable it after the backup/restore operation.

Database polling opens the database and monitors or "pings" the database. For the "`alter database archivelog`" command to succeed, make sure database polling is disabled and the database is mounted EXCLUSIVE before executing the command.

7.1.5 Failing Over a Cold Failover Cluster Database

In the failover operation, you need to fail over the virtual hostname and IP, and also the shared storage to the standby node. For details, refer to the instructions provided by your vendor clusterware.

7.2 Oracle Real Application Clusters Databases

You can also use an Oracle Real Application Clusters (Oracle RAC) database for the OracleAS Metadata Repository database. In an Oracle RAC database, you have multiple database instances running on different nodes and sharing access to an Oracle database. These database instances are linked by an interconnect.

The multiple database instances in an Oracle RAC configuration provide high availability through redundancy. An Oracle RAC configuration also provides scalability: you can simply add nodes to the cluster (for example, to handle increased traffic or to improve performance).

7.2.1 Installing an Oracle RAC Database

You need to use the Oracle database installer to install an Oracle RAC database. Refer to the Oracle RAC installation guide that is shipped with the database for details.

After you have installed an Oracle RAC database, you can load the OracleAS Metadata Repository into the Oracle RAC database using the OracleAS Metadata Repository Creation Assistant. See the *Oracle Application Server Metadata Repository Creation Assistant User's Guide* for details.

7.2.2 Running an Oracle RAC Database

For details on administering Oracle RAC databases, including performing procedures such as:

- stopping and starting database instances
- adding and deleting nodes and database instances
- managing storage
- managing backup and recovery
- troubleshooting

see the *Oracle Real Application Clusters Administrator's Guide*. You can find this guide in the Oracle Database documentation set.

7.2.3 Backing up an Oracle RAC Database

You back up an Oracle RAC database containing OracleAS Metadata Repository by following the normal backup procedures for any Oracle RAC database. See the *Oracle Real Application Clusters Administrator's Guide* for details.

7.3 Other High Availability Solutions for the OracleAS Metadata Repository Database

There are other types of solutions that provide high availability for the database. You can install the OracleAS Metadata Repository in such databases using the OracleAS Metadata Repository Creation Assistant. Examples of such solutions include storage snapshots, cloning, or local data guard. These solutions create a copy of the database from which you can perform a restore operation if the database fails. See the database documentation for details on these solutions.

7.4 Checking the Status of OracleAS Metadata Repository

To check the status of the OracleAS Metadata Repository database, run the following commands:

- Connect to the database and check its state:

```
ORACLE_HOME/bin/sqlplus /nolog
SQL> connect SYS as SYSDBA
SQL> select status from v$instance;
```

- Check the status of the database listener:

  ORACLE_HOME`/bin/lsnrctl status`

Part III

Oracle Internet Directory in High Availability Topologies

The chapters in this part provide details for running Oracle Internet Directory in high availability topologies.

This part contains the following chapters:

8

Oracle Internet Directory High Availability And Failover Considerations

While many Oracle customers deploy multiple identity management components, others choose to deploy only Oracle Internet Directory as a highly available identity repository. This chapter describes the availability and failover features of Oracle Internet Directory, and provides guidelines for exploiting these features in a typical directory deployment. It contains these topics:

8.1 About High Availability and Failover for Oracle Internet Directory

Oracle Internet Directory provides the high degree of system availability that mission-critical applications require. It does this by enabling:

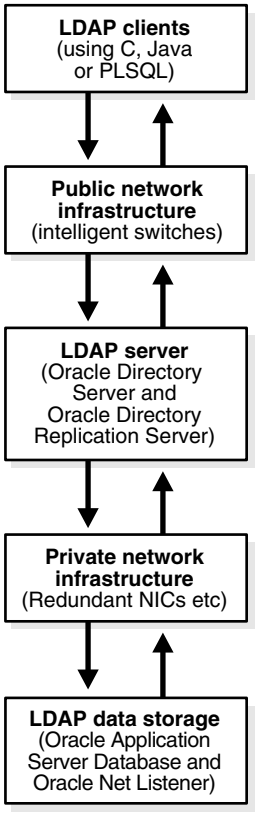
- All components in the system to facilitate redundancy
- All interfaces to facilitate failure recognition and recovery, called failover
- Integration of application-independent network failover capabilities in the overall deployment

Oracle products are commonly targeted for high availability environments and hence necessary capabilities are built into all layers of the Oracle technology stack. Typically, it is not necessary to employ every failover capability in every component.

8.2 Oracle Internet Directory and the Oracle Technology Stack

Figure 8–1 gives an overview of the various components of the Oracle Internet Directory stack. Stack communication between separate computers occurs by passing information from one node to the other through several layers of code. Information descends through layers on the client side. It is then packaged for transport across a network medium. The information then proceeds up the stack on the server side where it is translated and understood by the corresponding layers.

Figure 8–1 Oracle Internet Directory/Oracle Technology Stack



You can build sufficient fault tolerance mechanisms into each layer to ensure maximum availability of the product. The following sections describe some of the high availability options in each of these layers.

8.3 Failover Options on Clients

Incorporating enough intelligence in the clients so that they can failover to alternate Oracle directory servers in case the primary Oracle directory server fails is a good option in some cases. This requires the clients to cache alternate server information and use it upon recognizing connectivity loss. This method of guaranteeing availability is viable only for deployments in which one has full control over the type of clients accessing the directory.

This section contains these topics:

- Section 8.3.1, "Alternate Server List from User Input"
- Section 8.3.2, "Alternate Server List from the Oracle Internet Directory Server"

8.3.1 Alternate Server List from User Input

The clients can be designed to obtain the list of alternate Oracle directory servers from user input so that the clients can automatically failover in the event of a failure of the primary server. However, as the number of clients increases, this option does not scale very well in terms of administration of client installations.

8.3.2 Alternate Server List from the Oracle Internet Directory Server

Oracle Internet Directory supports a DSE root attribute called `AltServer`. This is an LDAP Version 3 standard attribute and is to be maintained by the directory administrator. It points to other Oracle directory servers in the system with the same set of naming contexts as that of the local server. When connectivity to the local server is lost, clients have the option of accessing one of the servers listed in this attribute. This option requires explicit administrative action to maintain this attribute.

Clients should cache the information in the alternate server list for use in the event that the primary server becomes unavailable.

8.3.2.1 Setting the Alternate Server List by Using Oracle Directory Manager

To set the alternate server list:

1. In the navigator pane, expand Oracle Internet Directory Servers, then select a server instance. System operational attributes appear in the right pane.
2. In the Alternate Server field, enter the name or names of alternate servers.
3. Choose OK.

See Also:

- RFC 2251 at `http://www.ietf.org` for details about the usage of `altServer` attribute
- "Managing Attributes by Using Command-Line Tools" in the "Directory Schema Administration" chapter in *Oracle Internet Directory Administrator's Guide* for instructions about setting the `AltServer` attribute

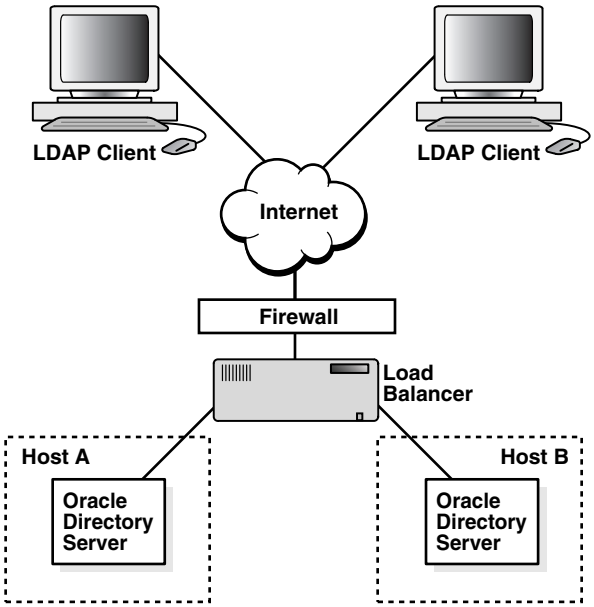
8.4 Failover Options in the Public Network Infrastructure

The network used to access Oracle Internet Directory services is called the Public Network Infrastructure. Providing network level load balancing and failover measures (connection re-direction) in the Public Network Infrastructure are highly recommended because these measures provide a high degree of flexibility and transparency to application clients.

If the Oracle Internet Directory services are accessed from the Internet, this would include a couple of high speed links (T1 to T3) and an intelligent TCP/IP level load balancer. If the Oracle Internet Directory services are accessed from an Intranet, this would include high speed LAN connections to the server computers running the Oracle directory server and an intelligent TCP/IP level load balancer. In both cases, there would be more than one computer serving LDAP requests so that failure of one Oracle directory server computer would not affect availability.

Figure 8–2 illustrates a typical Internet deployment of Oracle Internet Directory with network-level failover enabled.

Figure 8–2 Network-Level Failover



In Figure 8–2, the Oracle directory servers (LDAP servers) can be connected to either the same back-end database or different back-end databases. In this deployment, network-level load balancing can be accomplished by both hardware and software solutions.

This section contains these topics:

- Section 8.4.1, "Hardware-Based Load Balancing"
- Section 8.4.2, "Software-Based Load Balancing"

8.4.1 Hardware-Based Load Balancing

Hardware-based load balancing technology is available from several vendors. These redirection devices connect directly to the Internet and can route requests among several server computers. They can also detect computer failures and stop routing requests to the failed computer. This feature guarantees that new connections from clients will not be routed to a failed computer. When a computer comes back, the device detects it and starts routing new requests to it. These devices also perform some load balancing, which makes sure that client requests are uniformly distributed.

Some of the vendors providing hardware based re-direction technologies are:

- Accelar Server Switches from Nortel Networks
- Local Director from Cisco
- BIG/ip from F5 Labs Inc.
- Hydra from HydraWEB Technologies
- Equalizer from Coyote Point Systems

8.4.2 Software-Based Load Balancing

The software-based solutions essentially work in the same manner as their hardware counterparts. Some of the currently available solutions include Dispatch from Resonate and Network Dispatcher from IBM.

8.5 High Availability and Failover Capabilities in Oracle Internet Directory

Multimaster replication makes it possible for the directory system to be available for both access and updates at all times, as long as at least one of the nodes in the system is available. When a node comes back online after a period of unavailability, replication from the existing nodes will resume automatically and cause its contents to be synchronized transparently.

Any directory system with high availability requirements should always employ a network of replicated nodes in multimaster configuration. A replica node is recommended for each region that is separated from others by a relatively low speed or low bandwidth network segment. Such a configuration, while allowing speedy directory access to the clients in the same region, also serves as a failover arrangement during regional failures elsewhere.

8.6 Failover Options in the Private Network Infrastructure

The Private Network Infrastructure is the network used by Oracle Internet Directory and its back-end components to communicate with each other. In cases where Oracle Internet Directory is deployed on the Internet, Oracle Corporation recommends that this network be physically different from the network used to serve client requests. In cases where Oracle Internet Directory is deployed over an Intranet, the same LAN may be used, but Oracle Internet Directory components should have dedicated bandwidth with the help of a network switch. Because Oracle Internet Directory depends on the Private Network Infrastructure for its communications, you must take adequate precautions to guarantee availability in the event of failures in the Private Network. Some of the options available in this area are:

- Section 8.6.1, "IP Address Takeover (IPAT)"
- Section 8.6.2, "Redundant Links"

8.6.1 IP Address Takeover (IPAT)

IP address takeover feature is available on many commercial clusters. This feature protects an installation against failures of the Network Interface Cards (NICs). To make this mechanism work, installations must have two NICs for each IP address assigned to a server. Both the NICs must be connected to the same physical network. One NIC is always active while the other is in a standby mode. The moment the system detects a problem with the main adapter, it immediately fails over to the standby NIC. Ongoing TCP/IP connections are not disturbed and as a result clients do not notice any downtime on the server.

8.6.2 Redundant Links

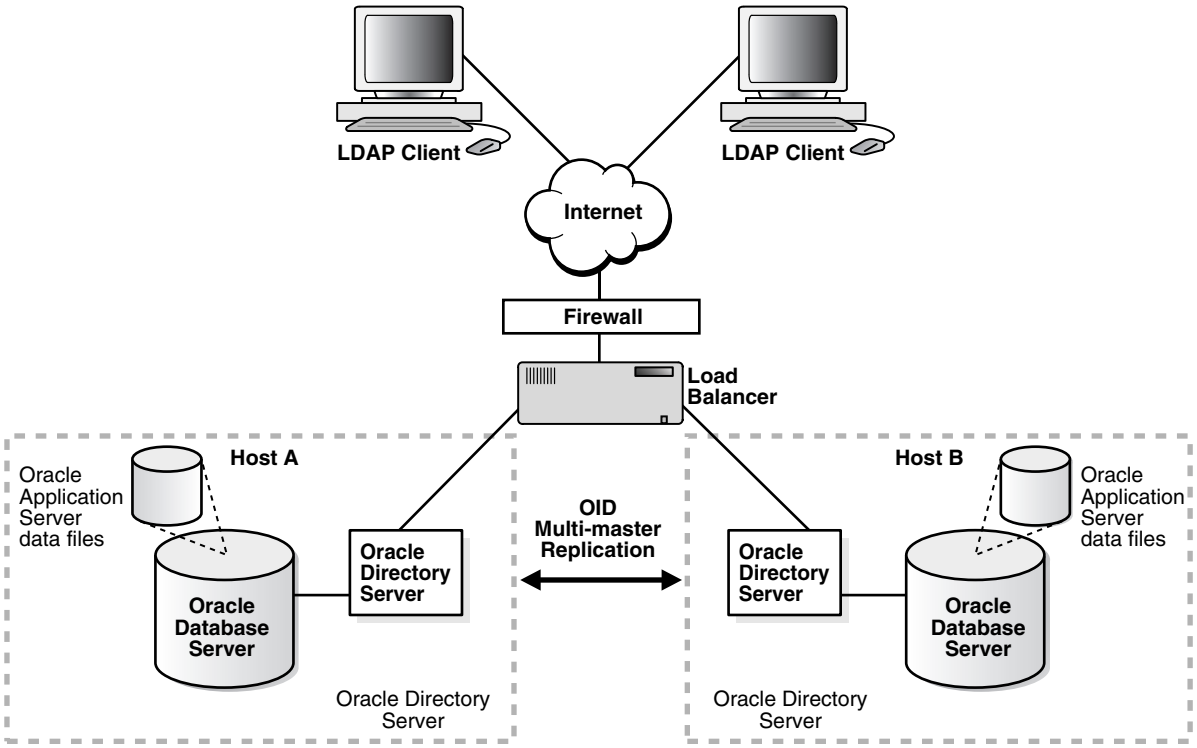
Since all networks (with the exception of wireless networks) are comprised of wires going from one location to the other, there is a distinct possibility that someone might unintentionally disconnect a wire that is used to link a client computer to a server computer. If you want to take such precautions, use NICs and hubs/switches that come with the capability to use redundant links in case of a link level failure.

8.7 High Availability Deployment Examples

In Figure 8–3, both the database and Oracle directory server (LDAP server) reside on the same computer. Changes on one directory server instance are reflected on the second directory server instance through multimaster replication. When a failure of

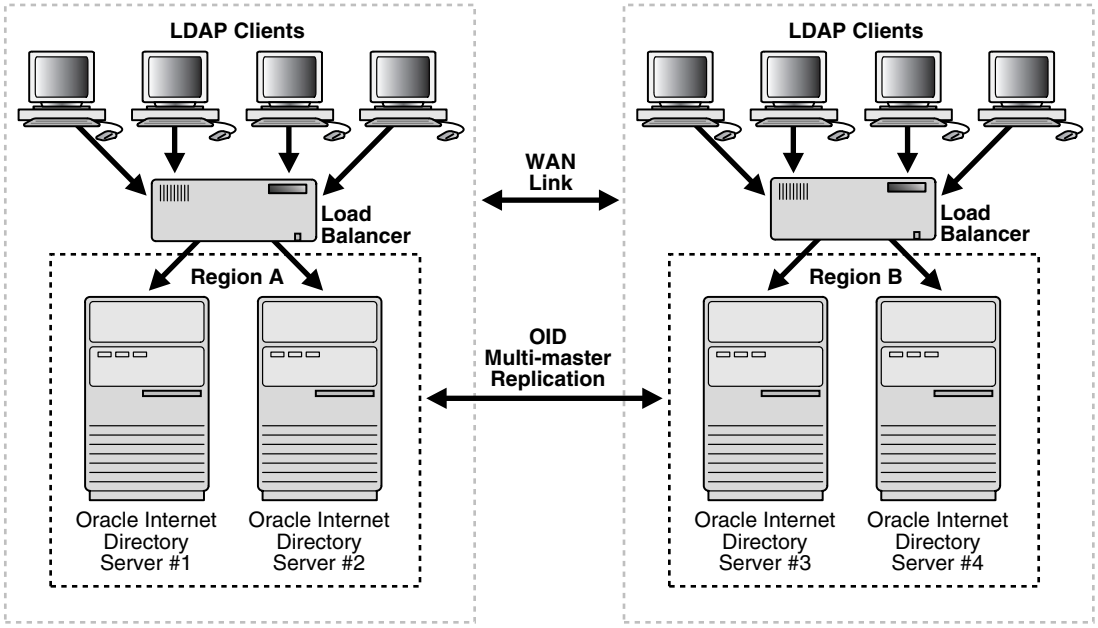
the directory server or database server on a particular node occurs, it is elevated to a computer failure so that the load balancer will stop handing off connections to the computer on which there was a failure.

Figure 8–3 Deployment Example (Two Oracle Internet Directory Nodes in Replication)



As Figure 8–4 illustrates, each region can be set up with two Oracle Internet Directory nodes replicating between each other. This configuration is typical of global directory networks deployed by large enterprises where each of the regions could potentially represent a continent or a country.

Figure 8–4 Deployment Example 2



9

Oracle Internet Directory in Oracle Real Application Clusters Environment

Oracle Real Application Clusters (Oracle RAC) is a computing environment that harnesses the processing power of multiple, interconnected computers. Along with a collection of hardware, called a cluster, it unites the processing power of each component to become a single, robust computing environment. A cluster comprises two or more computers, also called nodes.

This chapter discusses the ways you can run Oracle Internet Directory in an Oracle RAC system. It contains these topics:

9.1 Terminology

- Node

  A computer where an instance resides. It can be part of a Massively Parallel Computing Infrastructure in which it shares disk storage with other nodes. In most cases, a node has its own copy of the operating system.

- Cluster

  A set of instances, each typically running on a different node, that coordinate with each other when accessing the shared database on the disk

- Cluster Manager

  An operating system-dependent component that discovers and tracks the membership state of nodes by providing a common view of cluster membership across the cluster

- Transparent Application Failover (TAF)

  A runtime failover for high-availability environments, such as Oracle RAC and Oracle Fail Safe, that refers to the failover and re-establishment of

application-to-service connections. It allows client applications to automatically reconnect to the database if the connection fails, and optionally resume a SELECT statement that was in progress. This reconnect happens automatically from within the Oracle Call Interface (OCI).

The client notices no connection loss as long as there is one instance left serving the application.

- Connect-time failover

  Failover method in which a client connect request is forwarded to another listener if the first listener is not responding. It is enabled by service registration, because the listener knows whether an instance is running before attempting a connection.

9.2 Installing Oracle Internet Directory against an Oracle RAC Database

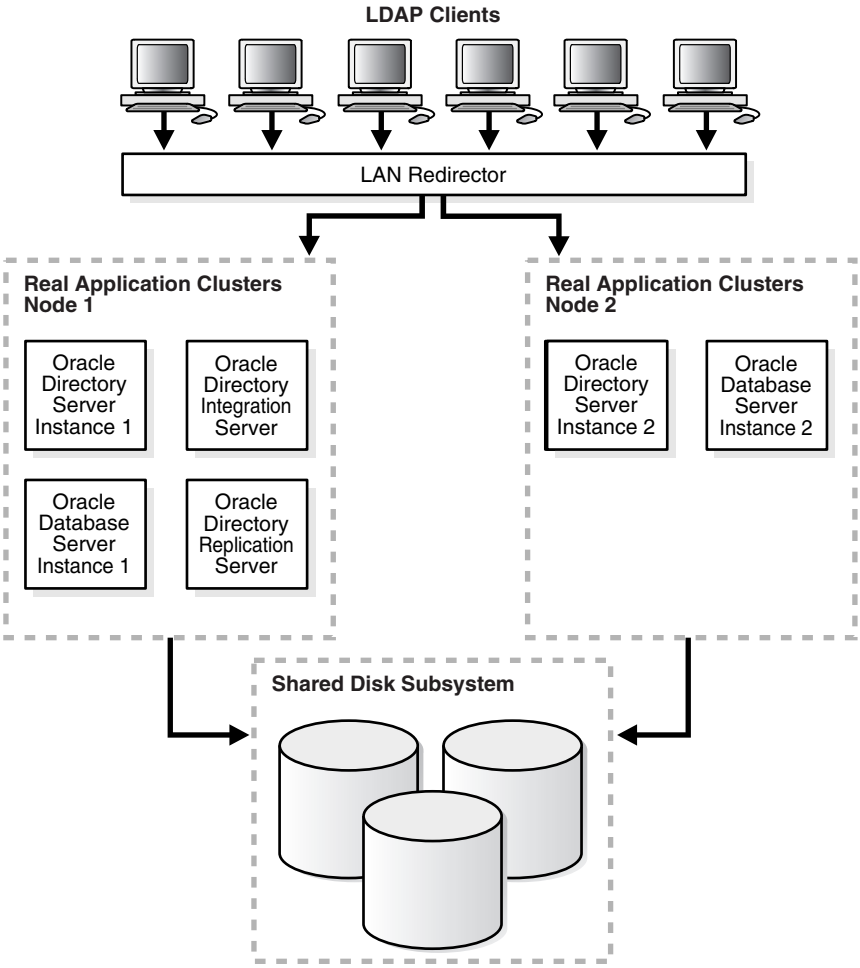
For information on installing Oracle Internet Directory against an Oracle RAC database, see the chapter entitled "Installing in High Availability Environments: OracleAS Cluster (Identity Management)" in the *Oracle Application Server Installation Guide*.

9.3 Oracle Internet Directory in an Oracle RAC Environment

To achieve a very comprehensive high availability configuration, you can configure Oracle Internet Directory to run on an Oracle RAC environment. This involves running Oracle Internet Directory processes and the Oracle Internet Directory-designated database on all the Oracle RAC nodes.

Figure 9–1 shows a two-node cluster on which an Oracle RAC database is configured.

Figure 9–1 Oracle Internet Directory with Basic High Availability Configuration



As Figure 9–1 shows:

- Oracle directory server instance 1 is active on Oracle RAC Node 1 and Oracle directory server instance 2 is active on Oracle RAC Node 2. Note that multiple Oracle directory server instances can be started on each node.
- Oracle Directory Integration Platform instances are active on both nodes.
- The Oracle directory replication server instance is active on one node only. If the node fails, then the OID Monitor on the surviving node pulls the Oracle directory replication server instance from the failed node and starts it on the surviving node.
- The LDAP client applications can be configured to communicate with Oracle Internet Directory on different Oracle RAC nodes directly. Alternatively, the Oracle Internet Directory server instances can be front-ended by a LAN redirector to get a single system image of the Oracle RAC nodes.
- When one Oracle RAC node is unavailable because of failure or maintenance purposes, Oracle Internet Directory running on the other Oracle RAC node is available. The LDAP clients connected to Oracle Internet Directory on the failed Oracle RAC node must reconnect.

9.4 Oracle Directory Server Connection Modes to Oracle RAC Database Instances

This section discusses the various connection modes possible for Oracle directory server instances communicating with Oracle RAC database instances. These connection modes are transparent to the Oracle Internet Directory clients, and do not affect the way in which Oracle Internet Directory communicates with its clients.

This section contains these topics:

9.4.1 Load_balance Parameter

If the `load_balance` parameter in the `tnsnames.ora` file is set to `ON`, then Oracle Internet Directory connections to the Oracle Database are distributed to each Oracle Database node. During failover of any node, only connections to the failed node are redirected to the available Oracle Database nodes.

If the `load_balance` parameter is set to `off`, then all the Oracle Internet Directory connections to the Oracle Database are to one Oracle Database node only.

During failover, all the connections are redirected to the available Oracle Database nodes.

9.4.2 Connect-Time Failover (CTF)

At the time of connection to the Oracle Database by the Oracle directory servers, if the primary Oracle Database node is not available, then Oracle Internet Directory servers connect to the backup (that is, secondary) database.

9.4.3 Transparent Application Failover (TAF)

To configure TAF, in the `tnsnames.ora` file, add one of the following:

- `type=select` and `method=preconnect`

or

- `type=select` and `method=basic`

During any LDAP search operation, if the primary Oracle Database node fails, then the Oracle directory server transparently connects to the backup (that is, the secondary) Oracle Database node, and the current LDAP search operation continues.

9.4.4 Configuring the tnsnames.ora File for the Failover

This section shows configurations of the `tnsnames.ora` files on two nodes.

Node 1

```
db.us.acme.com=
 (description=
  (load_balance=off/on)  /* only connect time load balancing & connection load
balancing */
  (failover=on)          /* only connect time failover */
```

```
  (address=
       (protocol=tcp)
       (host=db1)
       (port=1521))
 (address=
       (protocol=tcp)
       (host=db2)
       (port=1521))
  (connect_data=
     (service_name=db.us.acme.com)
     (failover_mode=
       (backup=db2.acme.com)
       (type=select)
       (method=preconnect))))

db2.acme.com=
 (description=
  (address=
       (protocol=tcp)
       (host=db2)
       (port=1521))
  (connect_data=
     (service_name=db.us.acme.com)
     (instance_name=db2)
     (failover_mode=
     (backup=db2.acme.com)
     (type=select)
     (method=preconnect))
     ))
```

Node 2

```
db.us.acme.com=
 (description=
  (load_balance=off/on)  /* only connect time load balancing & connection load
balancing */
  (failover=on)          /* only connect time failover */
  (address=
       (protocol=tcp)
       (host=db2)
       (port=1521))
 (address=
       (protocol=tcp)
       (host=db1)
       (port=1521))
  (connect_data=
     (service_name=db.us.acme.com)
     (failover_mode=
       (backup=db1.acme.com)
       (type=select)
       (method=preconnect))))

db1.acme.com=
 (description=
  (address=
       (protocol=tcp)
       (host=db1)
       (port=1521))
  (connect_data=
     (service_name=db.us.acme.com)
     (instance_name=db2)
```

```
(failover_mode=
(backup=db2.acme.com)
(type=select)
(method=preconnect))))
```

9.5 Oracle Directory Replication Between Oracle Internet Directory Oracle RAC Nodes

Directory replication can be configured between two or more Oracle Internet Directory Oracle RAC nodes.

- Each node in the directory replication group (DRG) is an Oracle Internet Directory Oracle RAC node.
- Directory replication brings in geographic availability, and the Oracle Internet Directory Oracle RAC nodes in the DRG ensure local availability, manageability, and scalability.

In the event that the Oracle directory replication server fails, or if the node running it fails, the OID Monitor starts the replication server on another node in the Oracle RAC. For details on how OID Monitor monitors the Oracle Internet Directory processes, see Section 3.7.2, "OID Monitor Details".

9.6 About Changing the ODS Password on an Oracle RAC System

If you change the `ODS` password on one Oracle RAC node by using the OID Database Password Utility (`oidpasswd`), then you must update the wallet `ORACLE_HOME/ldap/admin/oidpwdlldap1` on the other Oracle RAC nodes. Do this either by copying the changed wallet to all the nodes, or by invoking the OID Database Password Utility on all other nodes to update the wallet file only. This applies to the replication password changes also. Here the Replication Environment Management Tool is used instead of the OID Database Password Utility.

If you run the `oidpasswd` command on one node only, and do not update the wallet on all the Oracle RAC nodes, the OC4J_SECURITY instance will not be able to start on the other nodes. You will see this error in the `oidctl.log` file:

```
[gsdsiConnect] ORA-1017, ORA-01017: invalid username/password; logon denied.
```

The fix is to copy the `oidpwdlldap1` file to the other Oracle RAC nodes.

10

Deploying Identity Management with Multimaster Replication

This chapter provides high-level instructions for installing Oracle Identity Management components with Oracle Internet Directory multimaster replication. This chapter assumes that you are familiar with Oracle Application Server components, including: Oracle Internet Directory, OracleAS Single Sign-On, Oracle Delegated Administration Services, and Oracle Directory Integration Platform. You should also be familiar with Oracle Internet Directory replication concepts.

You might find the following documentation pointers useful:

For information on	See:
Running a replicated Oracle Internet Directory	■ "Oracle Internet Directory Replication Concepts" chapter in the *Oracle Internet Directory Administrator's Guide* ■ "Oracle Internet Directory Replication Installation and Configuration" chapter in the Oracle Internet Directory Administrator's Guide ■ "Oracle Internet Directory Replication Monitoring and Management" chapter in the *Oracle Internet Directory Administrator's Guide*
Deploying Oracle Identity Management with fan-out replication	*Oracle Identity Management Infrastructure Administrator's Guide*
Using Oracle Directory Integration Platform with Oracle Internet Directory	*Oracle Identity Management Integration Guide*
Using Oracle Delegated Administration Services with Oracle Internet Directory	*Oracle Identity Management Guide to Delegated Administration*

Keep the following points in mind when using the command-line tools mentioned in this chapter:

- The `ORACLE_HOME`, `MASTER_HOME`, and `REPLICA_HOME` variables designate absolute Oracle home paths.
- Use the appropriate path separator while running the commands. The notation in this chapter is based on the UNIX path variable notation. For example, the `ldapadd` tool is located in the `ORACLE_HOME/bin` directory in the UNIX environment. In the Windows environment, this tool is located in the `ORACLE_HOME\bin` directory.

- The PATH environment variable should include the ORACLE_HOME/bin, ORACLE_HOME/ldap/bin, and ORACLE_HOME/opmn/bin directories.
- Include $ORACLE_HOME/lib in the appropriate library environment variable. For example, in the Solaris environment, include $ORACLE_HOME/lib in the LD_LIBRARY_PATH environment variable.

This chapter contains the following sections:

10.1 Multimaster Identity Management Replication Configuration

Figure 10–1 shows an Oracle Application Server topology with Oracle Internet Directory and OracleAS Single Sign-On running in multimaster replication mode. The master Oracle Internet Directory instance runs on Host 1, the replica on Host 3. Each Oracle Internet Directory instance has its own database. A load balancer directs LDAP requests to the Oracle Internet Directory instances.

Host 2 and Host 4 run OracleAS Single Sign-On and Oracle Delegated Administration Services. On Host 2, these components use the Oracle Internet Directory on Host 1. Similarly, on Host 4, these components use the Oracle Internet Directory on Host 3. Note that these components access Oracle Internet Directory directly, without going through the load balancer. This means that OracleAS Single Sign-On and Oracle Delegated Administration Services on Host 2 always access the Oracle Internet Directory on Host 1, and the same components on Host 4 always access the Oracle Internet Directory on Host 3. You can think of Host 1 and Host 2 as one "stack", and Host 3 and Host 4 as another "stack".

Because OracleAS Single Sign-On and Oracle Delegated Administration Services on Host 2 and Host 4 use different metadata repositories, you cannot place them in the same OracleAS Cluster. Figure 10–1 shows them in different clusters. On Host 2, OracleAS Single Sign-On and Oracle Delegated Administration Services are in a cluster called dcmCluster1, and on Host 4, they are in a different cluster called dcmCluster2. In Figure 10–1 each cluster contains only one Oracle Application Server instance. You can add more instances to each cluster, if you want. See Section 10.1.10, "Installing Additional OracleAS Single Sign-On / Oracle Delegated Administration Services Instances in Each Replication Stack" for details.

A load balancer directs requests to OracleAS Single Sign-On and Oracle Delegated Administration Services on Host 2 and Host 4. The load balancer can direct requests to these components on either host (despite the fact that these components on each host use different Oracle Internet Directory instances and do not belong to the same cluster) because these components reference a replicated Oracle Internet Directory / OracleAS Single Sign-On environment, which keeps the Oracle Internet Directory-managed data in the two databases synchronized.

See Section 10.1.9, "Load Balancer Configuration in a Multimaster Replication Scenario" for details on configuring the load balancers in this topology.

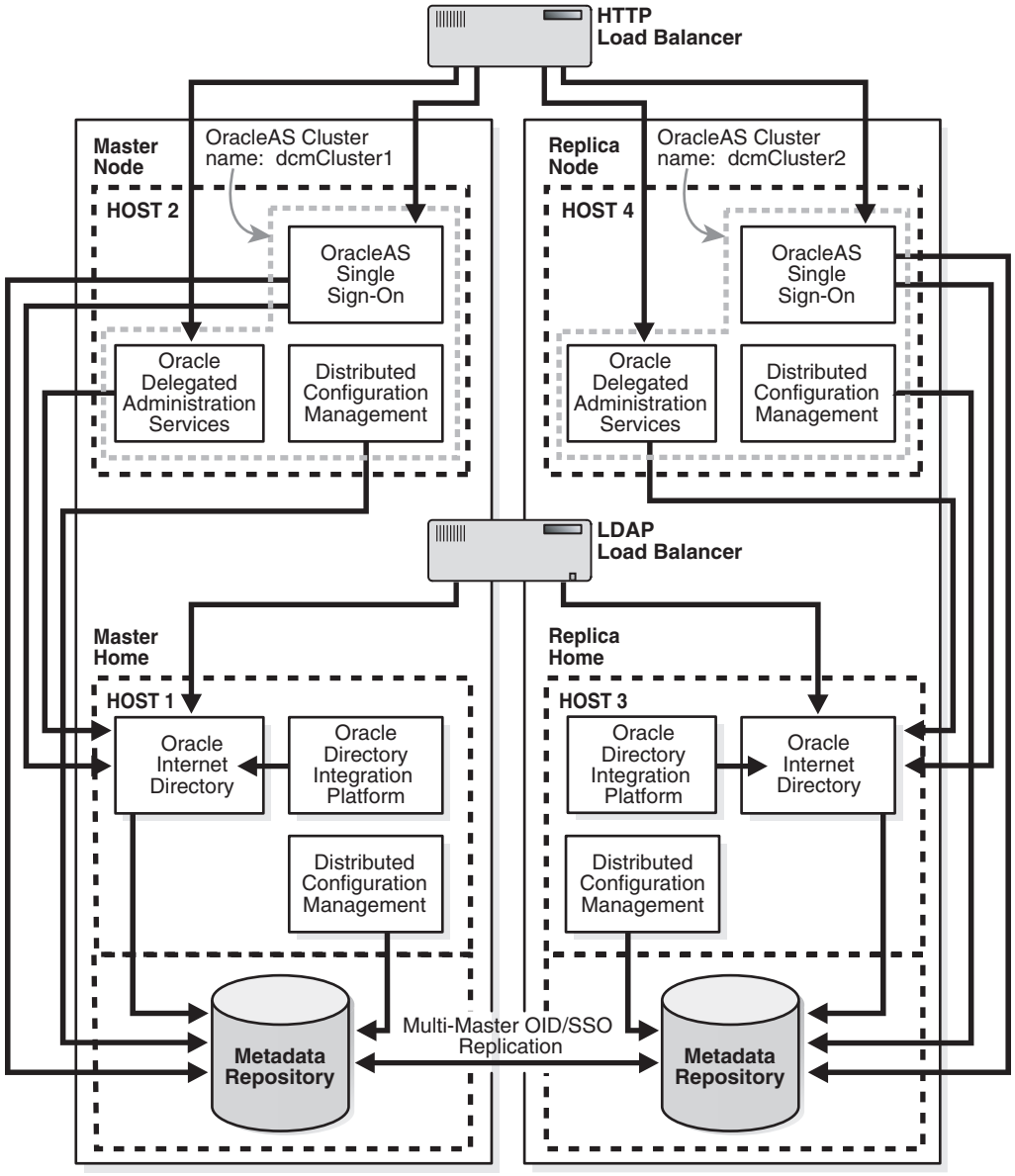
Steps to Create this Topology

To create this topology, perform these steps:

Table 10–1 Steps for Creating Multimaster Topology

Step	See this section
1. Install Oracle Internet Directory and Oracle Directory Integration Platform on the master node (Host 1).	Section 10.1.1, "Installing on the Master Node"
2. Install Oracle Internet Directory and Oracle Directory Integration Platform on the replica node (Host 3).	Section 10.1.2, "Installing on the Replica Node"
3. Configure Oracle Internet Directory for replication.	Section 10.1.3, "Setting up Multimaster Replication"
4. Install OracleAS Single Sign-On and Oracle Delegated Administration Services on Host 2.	Section 10.1.4, "Installing OracleAS Single Sign-On and Oracle Delegated Administration Services on the Master Node"
5. Synchronize the OracleAS Single Sign-On password.	Section 10.1.5, "Synchronizing the OracleAS Single Sign-On Schema Password"
6. Install OracleAS Single Sign-On and Oracle Delegated Administration Services on Host 4.	Section 10.1.6, "Installing OracleAS Single Sign-On and Oracle Delegated Administration Services on the Replica Node"

Figure 10–1 Multimaster Replication Topology



10.1.1 Installing on the Master Node

Install Oracle Internet Directory and Oracle Directory Integration Platform on the master node as follows:

- In the Oracle Application Server installer on Host 1: select Identity Management and Metadata Repository in the Select Installation Type screen, and select Oracle Internet Directory and Oracle Directory Integration Platform in the Select Configuration Options screen. This chapter refers to this Oracle home on Host 1 as the MASTER_HOME.
- Do not install any other Identity Management components such as OracleAS Single Sign-On or Oracle Delegated Administration Services on Host 1.

10.1.2 Installing on the Replica Node

Install Oracle Internet Directory with OracleAS Metadata Repository on the replica node as follows:

- In the Oracle Application Server installer on Host 3:

  Select Identity Management and Metadata Repository in the Select Installation Type screen.

  Select Oracle Internet Directory, Oracle Directory Integration Platform, High Availability and Replication in the Select Configuration Options screen.

  This chapter refers to this Oracle home on Host 3 as the REPLICA_HOME. This Oracle home will have only Oracle Internet Directory with OracleAS Metadata Repository and Oracle Directory Integration Platform. The OracleAS Metadata Repository database should have a unique global database name.

- Do not install any other Oracle Identity Management components, such as OracleAS Single Sign-On and Oracle Delegated Administration Services on Host 3.

> **Note:** When installing the replica, be sure to select **High Availability and Replication** in the Select Configuration Options screen so that the installer will prompt you for the replication type. It will ask you to select **ASR Replica** or **LDAP Replica**. Select **ASR Replica**.

10.1.3 Setting up Multimaster Replication

To set up the master and the replica nodes for replication, perform the following tasks described in the *Oracle Internet Directory Administrator's Guide*:

Item	Name
Book	*Oracle Internet Directory Administrator's Guide* This book is available in the Oracle Application Server documentation set.
Chapter	30, "Oracle Internet Directory Replication Installation and Configuration"
Section	30.3.2, "Installing and Configuring a Multimaster Replication Group"
Task	Task 3: Set Up Oracle Database Advanced Replication for a Directory Replication Group Task 5: Ensure that Oracle Directory Server Instances Are Started on All the Nodes Task 6: Start the Replication Servers on All Nodes in the DRG Task 7: Test Directory Replication

10.1.4 Installing OracleAS Single Sign-On and Oracle Delegated Administration Services on the Master Node

On the master node (Host 2 in Figure 10–1), install OracleAS Single Sign-On and Oracle Delegated Administration Services so that these components use the OracleAS Metadata Repository and Oracle Internet Directory on Host 1. To do this, make the following selections in the installation screens:

1. Specify File Locations - enter the destination directory where you want to install OracleAS Single Sign-On and Oracle Delegated Administration Services.
2. Select a Product to Install - select **Oracle Application Server Infrastructure**.
3. Select Installation Type - select **Identity Management**.
4. Confirm Pre-Installation Requirements - verify that you meet the requirements and select all the checkboxes.
5. Select Configuration Options - select **OracleAS Single Sign-On**, **Oracle Delegated Administration Services**, and **High Availability and Replication**.
6. Specify Port Configuration Options - select **Automatic**.
7. Select High Availability Option - select **OracleAS Cluster (Identity Management)**.
8. Create or Join an Oracle Application Server Cluster (Identity Management) - select **Create a New Oracle Application Server Cluster**.
9. Specify New Oracle Application Server Cluster Name - enter a name for the new cluster (for example: `dcmCluster1`).
10. Specify LDAP Virtual Host and Ports - enter the *physical hostname* of Host 1 (not the virtual name configured on the load balancer), and the necessary ports for Oracle Internet Directory.
11. Specify Oracle Internet Directory Login - enter the login and password for Oracle Internet Directory.
12. Specify HTTP Listen Port, Load Balancer Host and Port - enter the port number that you want to use for Oracle HTTP Server in **HTTP Listener Port**. In **HTTP Load Balancer Hostname** and **Port**, enter the HTTP virtual hostname configured on the load balancer and the port number configured for the virtual hostname.
13. Specify Instance Name and ias_admin Password - enter a name for this Oracle Application Server instance, and the password for the ias_admin user.

10.1.5 Synchronizing the OracleAS Single Sign-On Schema Password

To synchronize the OracleAS Single Sign-On schema password between the master Metadata Repository database (MDS) and the replica Metadata Repository database (RMS), follow the steps in the following section:

Item	Name
Book	*Oracle Application Server Single Sign-On Administrator's Guide* This book is available in the Oracle Application Server documentation set.
Chapter	9, "Advanced Deployment Options"
Section	9.2.2, "Configuring the Identity Management Database for Replication"
Step	Perform step 2.

Whenever you add a new OracleAS Single Sign-On and Oracle Delegated Administration Services replica, you must first perform this step from the master Oracle home on the replica to synchronize the OracleAS Single Sign-On schema password with the OracleAS Metadata Repository.

Note: If you encounter errors, the OracleAS Metadata Repository might be misconfigured. Either the MDS or RMS might not have the correct database information, as used by OracleAS Single Sign-On.

10.1.6 Installing OracleAS Single Sign-On and Oracle Delegated Administration Services on the Replica Node

Install OracleAS Single Sign-On and Oracle Delegated Administration Services on the replica node as follows:

1. On Host 4, install OracleAS Single Sign-On and Oracle Delegated Administration Services so that these components use the Metadata Repository and Oracle Internet Directory on the replica node (Host 3 in Figure 10–1). To do this, follow the screen sequence shown in Section 10.1.4, "Installing OracleAS Single Sign-On and Oracle Delegated Administration Services on the Master Node", with the following differences:
   - In step 8 on page 10-6, you also create a new cluster. You cannot join this instance (on Host 4) with the instance on Host 2 in the same cluster because the instances use different OracleAS Metadata Repositories.
   - In step 9 on page 10-6, enter a different cluster name (for example: `dcmCluster2`).
   - In step 10 on page 10-6, enter the physical hostname for Host 3 instead of Host 1, because you want OracleAS Single Sign-On and Oracle Delegated Administration Services to use the Oracle Internet Directory running on Host 3.
2. Synchronize the `mod_osso` configuration from the master middle tier, as described in the following section:

Item	Name
Book	*Oracle Application Server Single Sign-On Administrator's Guide* This book is available in the Oracle Application Server documentation set.
Chapter	9, "Advanced Deployment Options"
Section	9.1.2.3, "Configuration Steps"
Step	Reregister mod_osso on the single sign-on middle tiers

3. Repeat this procedure to install additional OracleAS Single Sign-On and Oracle Delegated Administration Services instances, as needed.

10.1.7 If You Are Running in SSL Mode

If you selected the SSL option when installing the OracleAS Single Sign-On and Oracle Delegated Administration Services components, be aware of the following situation that can cause an error when running in SSL mode.

When both nodes (Node 2 and Node 4 in Figure 10–1) running OracleAS Single Sign-On and Oracle Delegated Administration Services are active, you may see the following error in the browser when you try to access the OracleAS Single Sign-On Administrator page from the `/pls/orasso` URL:

```
Service Temporarily Unavailable
The server is temporarily unavailable to service your request due to
```

```
maintenance downtime or capacity problems. Please try again later.
```

In the `error_log` file, you may see the following error:

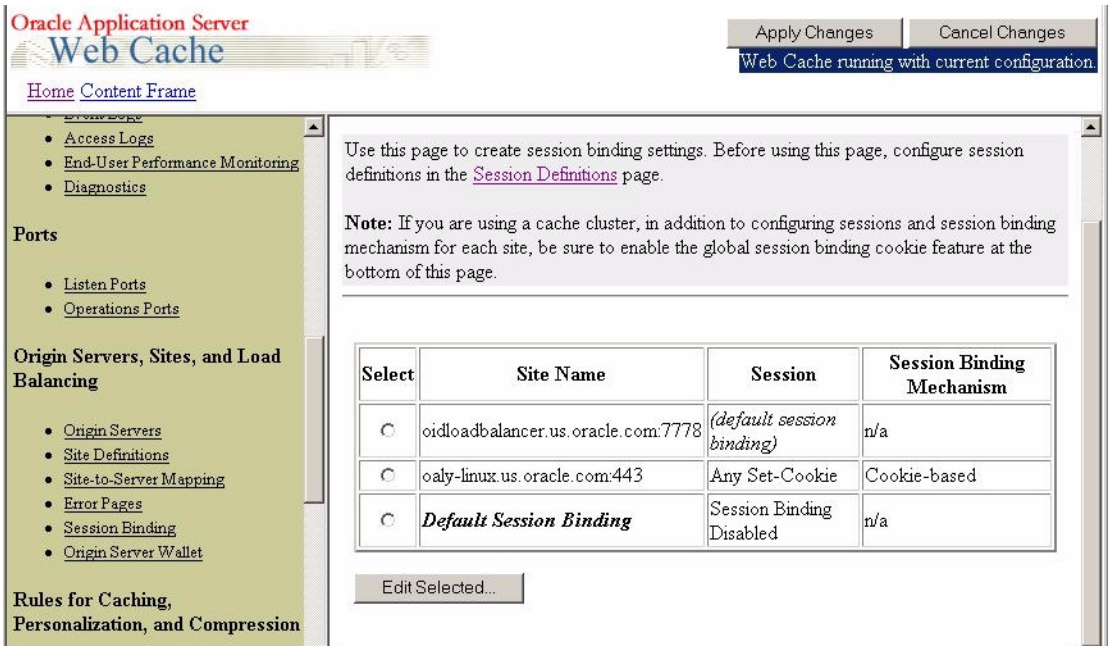
```
mod_plsql: /pls/orasso/ORASSO.home HTTP-503 ORA-20000   Call to WPG_SESSION
API Failed.
```

To fix this problem, check that your load balancer's persistence setting is configured correctly. If you are using OracleAS Web Cache as the load balancer, ensure that the session is set to "Any Set-Cookie", as shown in Figure 10–2.

If you are using a hardware load balancer, check the load balancer documentation and confirm that the load balancer's persistence setting is set to a value that supports SSL communication. If you are not sure which value to specify, contact the load balancer vendor directly.

For more information, see Oracle*MetaLink* Note 372956.1. You can access Oracle*MetaLink* at `http://metalink.oracle.com`.

Figure 10–2 In OracleAS Web Cache, Configure the Session to "Any Set-Cookie"



10.1.8 Oracle Directory Integration Platform Event Propagation in a Multimaster Scenario

Oracle Directory Integration Platform supports high availability in an Oracle Internet Directory multimaster replicated scenario, with certain drawbacks. In this high availability scenario, when changes are applied to Oracle Internet Directory on one node, the changes get propagated to the other consumer nodes. The Oracle Directory Integration Platform server running on each node is responsible for event propagation to the configured applications on that node. That is, the applications that have provisioning profiles on that Oracle Internet Directory node will be informed of the changes happening on that Oracle Internet Directory node.

See Also: *Oracle Identity Management Integration Guide*

10.1.9 Load Balancer Configuration in a Multimaster Replication Scenario

Figure 10–1 shows two load balancers: one for HTTP requests and one for LDAP requests. Note the following points when you configure these load balancers:

- The LDAP load balancer does not accept requests from OracleAS Single Sign-On and Oracle Delegated Administration Services.

  OracleAS Single Sign-On and Oracle Delegated Administration Services should not use the LDAP load balancer because they need to send requests only to the Oracle Internet Directory *in the same "stack"*, where a stack consists of OracleAS Single Sign-On and its corresponding Oracle Internet Directory. You associated this OracleAS Single Sign-On with its Oracle Internet Directory during installation (see step 10 on page 10-6).

  For example, in Figure 10–1, OracleAS Single Sign-On and Oracle Delegated Administration Services on Host 2 and the Oracle Internet Directory on Host 1 make up one stack, and OracleAS Single Sign-On and Oracle Delegated Administration Services on Host 4 and the Oracle Internet Directory on Host 3 make up another stack.

- All other LDAP requests (other than the ones from OracleAS Single Sign-On / Oracle Delegated Administration Services) should go through the LDAP load balancer. For example, requests from OracleAS Portal should go through the LDAP load balancer.

- The HTTP load balancer should monitor both the OracleAS Single Sign-On servers and the Oracle Internet Directory servers on all nodes. It needs to do this so that it can ensure that the HTTP and LDAP requests are routed to the same "stack". For example, if the Oracle Internet Directory on Host 1 is down, then the HTTP load balancer should route HTTP requests only to the OracleAS Single Sign-On server on Host 4 because its Oracle Internet Directory server on Host 3 is up.

- The HTTP load balancer should be configured for persistent routing of HTTP requests.

For details on deploying applications in a replicated environment, see section 3.3.2.7, "Application Deployments in Replicated Directory Environments", in the *Oracle Identity Management Infrastructure Administrator's Guide*.

10.1.10 Installing Additional OracleAS Single Sign-On / Oracle Delegated Administration Services Instances in Each Replication Stack

You can add OracleAS Single Sign-On and Oracle Delegated Administration Services instances to each of the OracleAS Clusters that you created previously in Section 10.1.4 and Section 10.1.6. Each cluster will provide redundancy for OracleAS Single Sign-On and Oracle Delegated Administration Services in each replica's stack.

To do this, follow the generic indications and recommendations for Distributed OracleAS Cluster (Identity Management) installations as described in the *Oracle Application Server Installation Guide*, and make the following selections in the installation screens:

1. Specify File Locations - enter the destination directory where you want to install OracleAS Single Sign-On and Oracle Delegated Administration Services.
2. Select a Product to Install - select **Oracle Application Server Infrastructure**.
3. Select Installation Type - select **Identity Management**.

4. Confirm Pre-Installation Requirements - verify that you meet the requirements and select all the checkboxes.
5. Select Configuration Options - select **OracleAS Single Sign-On**, **Oracle Delegated Administration Services**, and **High Availability and Replication**.
6. Specify Port Configuration Options - select **Automatic**.
7. Select High Availability Option - select **OracleAS Cluster (Identity Management)**.
8. Create or Join an Oracle Application Server Cluster (Identity Management) - select **Join an Oracle Application Server Cluster**.
9. Specify Existing OracleAS Cluster Name - enter the name of the cluster (for example: `dcmCluster1` for the master (Host 1/Host 2) stack, or `dcmCluster2` for the replica (Host 3/Host 4) stack).
10. Specify LDAP Virtual Host and Ports - enter the *physical hostname* of Host 1 (not the virtual name configured on the load balancer), and the necessary ports for Oracle Internet Directory. If you are installing on the replica stack, enter the *physical hostname* of Host 3.
11. Specify Oracle Internet Directory Login - enter the login and password for Oracle Internet Directory.
12. Specify HTTP Listen Port, Load Balancer Host and Port - enter the port number that you want to use for Oracle HTTP Server in **HTTP Listener Port**. In **HTTP Load Balancer Hostname** and **Port**, enter the HTTP virtual hostname configured on the load balancer and the port number configured for the virtual hostname.
13. Specify Instance Name and ias_admin Password - enter a name for this Oracle Application Server instance, and the password for the ias_admin user.

10.2 Adding a Node to a Multimaster Replication Group

To add a replication node to a functioning directory replication group (DRG), follow these steps.

1. First, install the new node.

   Install Identity Management and Metadata Repository. This installation will have only the Metadata Repository, Oracle Internet Directory and Oracle Directory Integration Platform. The replica node Metadata Repository should have a unique global database name.

   Do not install other Identity Management components such as OracleAS Single Sign-On or Oracle Delegated Administration Services.

2. Prepare the environment for adding a node.

   a. Configure the Oracle Net Services environment as described in Task 3, Installing and Configuring a Multimaster Replication Group, in the "Oracle Internet Directory Replication Administration" chapter of *Oracle Internet Directory Administrator's Guide*.

   b. Stop the directory replication server on all nodes

   c. Identify a sponsor node and switch the sponsor node to read-only mode

      Note: While the sponsor node is in read-only mode, do not make any updates to it. You may, however, update any of the other nodes, but those updates are not replicated immediately. Also, the sponsor node and the MDS can be the same node.

d. Back up the sponsor node by using `ldifwrite`. Enter the following command:

```
ORACLE_HOME/bin/ldifwrite -c connect_string  \
        -b "orclagreementid=000001,cn=replication configuration" \
        -f output_ldif_file
```

3. Add the node into the replication group.

a. Perform the Advanced Replication add node setup on the sponsor node by typing:

```
ORACLE_HOME/bin/remtool -addnode
```

The Replication Environment Management Tool adds the node to the DRG.

Note: Note: If you encounter errors, then use `remtool -asrverify`. If it reports errors, then rectify them by using `remtool -asrrectify`. Both of those options list all the nodes in the DRG. If the node to be deleted is in the list, then delete it by running `remtool -delnode` again.

---

b. Switch the sponsor node to updatable mode.

c. Start the directory replication server on all nodes except the new node.

d. Stop `oidmon`

e. Load data into the new node, as follows:

First do a check and generate by typing:

```
ORACLE_HOME/ldap/bin/bulkload.sh \
  -connect <db_connect_string_of_new_node> \
  -check -generate -restore  \
  absolute_path_to_the_ldif_file_generated_by_ldifwrite
```

Note: Verify that the `ORACLE_HOME/ldap/log/bulkload.log` does not report any errors. It is possible that you might see `Duplicate entry errors` in the log for some of the entries. You can safely ignore this error and proceed with the load.

Now load the data on the target node by typing:

```
ORACLE_HOME/ldap/bin/bulkload.sh \
  -connect db_connect_string_of_new_node \
  -load -restore  \
  absolute_path_to_the_ldif_file_generated_by_ldifwrite
```

4. Start the directory server on the new node by typing the following command:

```
ORACLE_HOME/opmn/bin/opmnctl startproc ias-component=OID
```

5. Start the directory replication server on the new node by typing:

```
ORACLE_HOME/bin/oidctl connect=db_connect_string_of_new_node \
   server=oidrepld instance=1 \
   flags='-h host_name_of_new_node -p port'  start
```

6. Install a new middle tier, based on the new replica node.
   a. Synchronize the OracleAS Single Sign-On schema passwords from MDS to the new node as described in Section 10.1.5, "Synchronizing the OracleAS Single Sign-On Schema Password".
   b. Install OracleAS Single Sign-On and Oracle Delegated Administration Services as described in Section 10.1.6, "Installing OracleAS Single Sign-On and Oracle Delegated Administration Services on the Replica Node".
   c. Configure the HTTP load balancer to distribute incoming traffic to this newly installed node.

10.3 Deleting a Node from a Multimaster Replication Group

You can delete a node from a DRG, provided the DRG contains more than two nodes. You might need to do so if the addition of a new node did not fully succeed as a result of system errors. To delete a replication node, perform these steps:

1. Stop the directory replication server on all nodes. To do that, run the following command on each node in the DRG:

```
ORACLE_HOME/bin/oidctl connect=connect_string server=oidrepld instance=1 stop
```

> **Note:** The instance number may vary.

2. Stop all processes on the node to be deleted.
   a. Stop all processes in the associated middle tier Oracle homes.

```
ORACLE_HOME/opmn/bin/opmnctl stopall
```

   b. On the node to be deleted, stop all Oracle Application Server processes including Oracle Internet Directory Monitor and all directory server instances.

```
ORACLE_HOME/opmn/bin/opmnctl stopall
```

3. Delete the node from the master definition site. From the MDS, run the following command:

```
ORACLE_HOME/bin/remtool -delnode
```

> **Note:** If you encounter errors, then use `remtool -asrverify`. If it reports errors, then rectify them by using `remtool -asrrectify`. Both of those options list all nodes in the DRG. If the new node is not in the list, then add it by running `remtool -addnode` again.

4. Start the directory replication server on all nodes by typing the following command:

```
ORACLE_HOME/bin/oidctl connect=connect_string server=oidrepld \
  instance=1 flags='-h host -p port' start
```

5. Decommission the removed node and its associated middle tier. You can optionally decommission the removed replicated node and associated middle tier by deinstalling the corresponding Oracle homes.

Part IV

Disaster Recovery

The chapter in this part describes the Oracle Application Server Disaster Recovery solution.

This part contains the following chapter:

11

OracleAS Disaster Recovery

Disaster recovery refers to how a system recovers from catastrophic site failures caused by natural or unnatural disasters. Examples of catastrophic failures include earthquakes, tornadoes, floods, or fire. Additionally, disaster recovery can also refer to how a system is managed for planned outages. For most disaster recovery situations, the solution involves replicating an entire site, not just pieces of hardware or subcomponents. This also applies to the Oracle Application Server Disaster Recovery (OracleAS Disaster Recovery) solution.

This chapter describes the OracleAS Disaster Recovery solution, how to configure and set up its environment, and how to manage the solution for high availability. The discussion involves both OracleAS middle tiers and OracleAS Infrastructure tiers in two sites: production and standby. The standby site is configured either identically and symmetrically or asymmetrically to the production site. Under normal operation, the production site actively services requests. The standby site is maintained to mirror or closely mirror the applications and content hosted by the production site.

The sites are managed using Oracle Application Server Guard, which contains a command-line utility (asgctl) that encapsulates administrative tasks (see Chapter 12, "OracleAS Guard asgctl Command-line Reference" for reference information about these administrative commands). The OracleAS Disaster Recovery solution leverages the following services among other system services that are available across the entire site. Behind the scenes OracleAS Guard automates the use of Backup and Recovery Tool (for managing configuration files in the file system) and Oracle Data Guard (for managing the OracleAS Infrastructure database) in a distributed fashion across the topology. Table 11–1 provides a summary of the OracleAS Disaster Recovery strategy and how this Oracle software is used behind the scenes:

Table 11–1 Overview of OracleAS Disaster Recovery strategy

Coverage	Procedure	Purpose
Middle-tier Configuration Files	OracleAS Backup and Recovery Tool	To back up OracleAS configuration files in the production site middle-tier nodes and restore the files to the standby site middle-tier nodes.
OracleAS Infrastructure Configuration Files	OracleAS Backup and Recovery Tool	To back up OracleAS configuration files in the production site OracleAS Infrastructure node and restore them to the standby site OracleAS Infrastructure node.
OracleAS Infrastructure Database	Oracle Data Guard	To ship archive logs from production site OracleAS Infrastructure database to standby site OracleAS Infrastructure database. Logs are not applied immediately.

Beginning with OracleAS release 10.1.2.0.2, to simplify the concepts presented to describe the OracleAS Disaster Recovery solution, the term topology is introduced to mean all farms on either the production or standby site. The term topology replaces the previous concept of a farm as described in the OracleAS Disaster Recovery solution documentation for OracleAS release 10.1.2.0.0. The term topology refers to all instances that share the same Oracle Internet Directory for a production site. The discover topology command queries Oracle Internet Directory to determine the list of instances and then generates a topology XML file that describes the production topology. The discover topology within farm command is used in cases where Oracle Internet Directory is not available and then OracleAS Guard uses OPMN to discover the topology within the farm.

Note: Your other databases must be covered in the overall disaster recovery strategy, and you must use Oracle Data Guard as the solution.

In addition to the recovery strategies, configuration and installation of both sites are discussed. For these tasks, two different ways of naming the middle-tier nodes are covered as well as two ways of resolving hostnames intra-site and inter-site.

With OracleAS Disaster Recovery, planned outages of the production site can be performed without interruption of service by switching over to the standby site using the OracleAS Guard switchover operation. Unplanned outages are managed by failing over to the standby site using the OracleAS Guard failover operation. Procedures for switchover and failover are covered in this chapter in Section 11.10, "Runtime Operations -- OracleAS Guard Switchover and Failover Operations".

This chapter is organized into the following sections:

- Section 11.1, "Oracle Application Server 10g Disaster Recovery Solution"
- Section 11.2, "Preparing the OracleAS Disaster Recovery Environment"
- Section 11.3, "Overview of Installing Oracle Application Server"
- Section 11.4, "Overview of OracleAS Guard and asgctl"
- Section 11.5, "Authentication of Databases"
- Section 11.6, "Discovering, Dumping, and Verifying the Topology"
- Section 11.7, "Dumping Policy Files and Using Policy Files With Some asgctl Commands"
- Section 11.8, "OracleAS Guard Operations -- Standby Site Cloning of One or More Production Instances to a Standby System"
- Section 11.9, "OracleAS Guard Operations -- Standby Instantiation and Standby Synchronization"
- Section 11.10, "Runtime Operations -- OracleAS Guard Switchover and Failover Operations"
- Section 11.11, "Monitoring OracleAS Guard Operations and Troubleshooting"
- Section 11.12, "Wide Area DNS Operations"
- Section 11.13, "Using OracleAS Guard Command-Line Utility (asgctl)"
- Section 11.14, "Special Considerations for Some OracleAS Metadata Repository Configurations"

- Section 11.15, "Special Considerations for OracleAS Disaster Recovery Environments"

> **See Also:** *Oracle Application Server Installation Guide* for instructions about how to install the OracleAS Disaster Recovery solution.

Geographically distributed Identity Management (IM) Infrastructure deployment replication, though an example of an active-active configuration, shares some features similar to an OracleAS Disaster Recovery solution in that Oracle Internet Directory (OID), OracleAS Metadata Repository (MR), and OracleAS Single Sign-On (SSO) are set up in replication and distributed across different geographic regions. Each OracleAS Single Sign-On site uses its own Oracle Internet Directory and OracleAS Metadata Repository located at the local site, thus resulting in the active-active configuration. The shared similarities serve two purposes. First, in case a database failure is detected at one site, Oracle Internet Directory and OracleAS Single Sign-On servers are reconfigured to route user requests to the closest geographic area. Second, in case a OracleAS Single Sign-On middle-tier failure is detected, the network is reconfigured to route traffic to a remote middle tier. However, this solution does not provide synchronization for OracleAS Portal, OracleAS Wireless, and Distributed Configuration Management (DCM) schemas in the Infrastructure database because neither supports the replica model used for Oracle Internet Directory and OracleAS Single Sign-On information. See *Oracle Identity Management Concepts and Deployment Planning Guide* for more information about a geographically distributed Identity Management Infrastructure deployment.

11.1 Oracle Application Server 10*g* Disaster Recovery Solution

The Oracle Application Server Disaster Recovery solution consists of two configured sites - one primary (production/active) and one secondary (standby). Both sites may or may not have the following: same number of middle tiers and the same number of OracleAS Infrastructure nodes, and the same number and types of components installed. In other words, the installations on both sites, middle tier and OracleAS Infrastructure could be identical (symmetrical topology) or not identical (asymmetrical topology). Both sites are usually dispersed geographically, and if so, they are connected through a wide area network.

Some important points to emphasize for the Oracle Application Server Disaster Recovery solution are the following:

- The number of instances required on the standby site to run your site can be identical to (symmetric) or fewer (asymmetric) than the production site.
- The set of instances needed must be created and installed on the standby site in case of failover.
- The standby site needs the minimum set of instances required to run your site.

This section describes the overall layout of the solution, the major components involved, and the configuration of these components. It has the following sections:

- Section 11.1.1, "OracleAS Disaster Recovery Requirements"
- Section 11.1.2, "Supported Oracle Application Server Releases and Operating Systems"
- Section 11.1.3, "Supported Topologies"

11.1.1 OracleAS Disaster Recovery Requirements

To ensure that your implementation of the OracleAS Disaster Recovery solution performs as designed, the following requirements must be adhered to:

- On each host in the standby site, make sure the following is identical to its equivalent peer in the production site:
  - For the middle-tier hosts, physical hostnames.

    Note: If you already have installed systems, you only need to modify the physical names for the middle-tier systems at the standby site and then create a virtual hostname for the physical hostname of the OracleAS Infrastructure (see the next bullet). See Section 11.2.1, "Planning and Assigning Hostnames" for information about how to change these physical hostnames and the virtual hostname.

  - Virtual hostname for the OracleAS Infrastructure. The virtual hostname can be specified in the **Specify Virtual Hostname** screen presented by the installer.
  - Hardware platform
  - Operating system release and patch levels
- All installations conform to the requirements listed in the *Oracle Application Server Installation Guide* to install Oracle Application Server.
- Oracle Application Server software is installed in identical directory paths between each host in the production site and its equivalent peer in the standby site.
- The following details must be the same between a host in the production site and a peer in the standby site:
  - User name and password of the user who installed Oracle Application Server must be the same between a host in the production site and its peer in the standby site.
  - Numerical user ID of the user who installed Oracle Application Server on that particular node
  - Group name of the user who installed Oracle Application Server on that particular node
  - Numerical group ID of the group for the user who installed Oracle Application Server on that particular node
  - Environment profile
  - Shell (command-line environment)
  - Directory structure, Oracle home names, and path of the Oracle home directory for each OracleAS installation on a node. Do not use symbolic links anywhere in the path.
  - Oracle Application Server installation types (Any instance installed on the standby system must be identical to that installed on the production system):
    * Middle Tier: J2EE and Web Cache, and Portal and Wireless
    * OracleAS Infrastructure: Metadata Repository (MR) and Identity Management (IM)

11.1.2 Supported Oracle Application Server Releases and Operating Systems

OracleAS Guard supports Oracle Application Server releases 10g (9.0.4) and 10g (10.1.2.0.0 and 10.1.2.0.2). The OracleAS Guard kit located on the 10g (10.1.2.0.2) Utility media #2 must be installed on all systems with Oracle homes or Oracle Application Server instances in the topology. See the OracleAS Disaster Recovery installation information in *Oracle Application Server Installation Guide* for more information.

OracleAS Guard supports a mixed OracleAS 10g (9.0.4) and OracleAS 10g (10.1.2) or higher release environment, such as may occur during an upgrade scenario. In this case, you may have upgraded the standby site to OracleAS 10g (10.1.2) Oracle homes in the middle tier but the Infrastructure is still an OracleAS 10g (9.0.4) Infrastructure. This mixed release environment will work as long as the OracleAS Disaster Recovery peer home for the given production middle tier Oracle homes are also upgraded to the same release OracleAS 10g (10.1.2) and as long as its Infrastructure is still an OracleAS 10g (9.0.4) Infrastructure. So the rule of thumb is that all Oracle home peer middle tiers within the topology must match exactly in release number on both the standby and production sites. Also the Infrastructures must match exactly in release number on both the standby and production sites and Oracle AS Guard must be the same version on both sites. However, as an upgrade based requirement, the Infrastructure can be at a lower release than the middle tiers because this happens during an upgrade scenario.

11.1.3 Supported Topologies

OracleAS Disaster Recovery supports a number of basic topologies for the configuration of the Infrastructure and middle tier on production and standby sites. OracleAS Disaster Recovery supports these basic topologies:

- Symmetrical Topologies - Strict Mirror of the Production Site with Collocated Oracle Identity Management and OracleAS Metadata Repository Infrastructure
- Asymmetrical Topologies - Simple Asymmetric Standby Topology with Collocated Oracle Identity Management and OracleAS Metadata Repository Infrastructure
- Separate OracleAS Metadata Repository for OracleAS Portal with Collocated Oracle Identity Management and OracleAS Metadata Repository Infrastructure (the Departmental Topology)
- Distributed Application OracleAS Metadata Repositories with Non Collocated Oracle Identity Management and OracleAS Metadata Repository Infrastructure

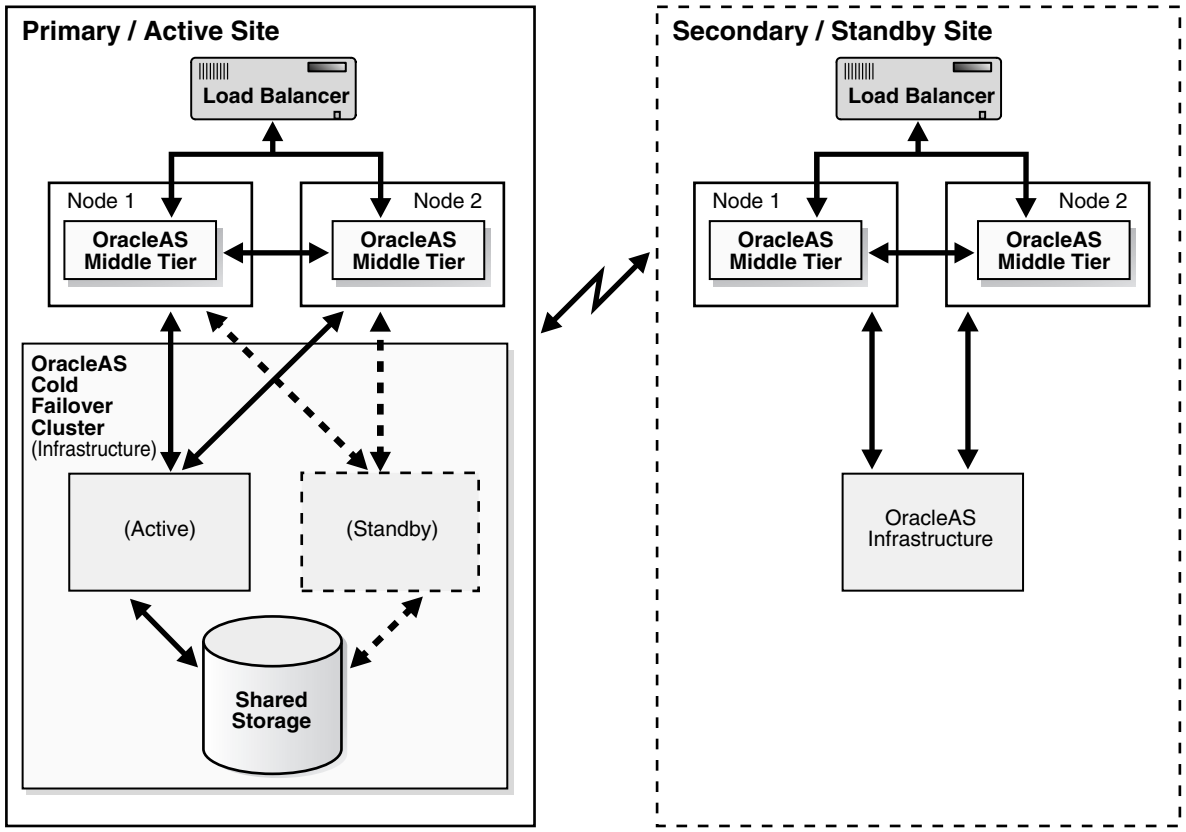
11.1.3.1 Symmetrical Topologies - Strict Mirror of the Production Site with Collocated Oracle Identity Management and OracleAS Metadata Repository Infrastructure

For OracleAS Disaster Recovery Release 10.1.2.0.1, only the OracleAS Disaster Recovery symmetrical topology environment was supported. This OracleAS Disaster Recovery environment has two major requirements:

- The deployment must use a single default Infrastructure install that contains a collocated OracleAS Metadata Repository and Oracle Identity Management.
- The standby site has to be a strict mirror of the production site with the same number of instances (symmetrical topology).

Figure 11–1 depicts an example OracleAS Disaster Recovery solution having a symmetrical topology with a Cold Failover Cluster on the primary site. This is considered a symmetrical topology because from an Oracle Application Server perspective both sites contain two OracleAS middle tiers and one Infrastructure.

Figure 11–1 Example Oracle Application Server Site-to-Site Disaster Recovery Solution (Load Balancer Appliance Is Optional If Only One Middle-Tier Node Is Present)



The procedures and steps for configuring and operating the OracleAS Disaster Recovery solution support 1 to *n* number of middle-tier installations in the production site. The same number of middle-tier installations must exist in the standby site. The middle tiers must mirror each other in the production and standby sites.

For the OracleAS Infrastructure, a uniform number of installations is not required (names or instances must be equal) between the production and standby sites. For example, the OracleAS Cold Failover Cluster (Infrastructure) solution can be deployed in the production site, and a single node installation of the OracleAS Infrastructure can be deployed in the standby site as shown in Figure 11–1. This way, the production site's OracleAS Infrastructure has protection from host failure using an OracleAS Cold Failover Cluster. This solution provides hardware redundancy by utilizing a virtual hostname. Refer to the section Section 4.2, "Common Characteristics of OracleAS Cold Failover Cluster Topologies" on page 4-12 for more information on OracleAS Cold Failover Clusters.

The OracleAS Disaster Recovery solution is an extension to various single-site Oracle Application Server architectures. Examples of such single-site architectures include the combination of OracleAS Cold Failover Cluster (Infrastructure) and active-active Oracle Application Server middle-tier architecture. For the latest information on what single-site architectures are supported, check the Oracle Technology Network (OTN) Web site for the latest certification matrix.

`http://www.oracle.com/technology/products/ias/hi_av/index.html`

The following are important characteristics of the OracleAS Disaster Recovery solution:

- Middle-tier installations are identical between the production and standby sites. In other words, each middle-tier installation in the production site has an identical installation in the standby site. More than one middle-tier node is recommended because this enables each set of middle-tier installations on each site to be redundant. Because they are on multiple machines, problems and outages within a site of middle-tier installations are transparent to clients.
- The OracleAS Disaster Recovery solution is restricted to identical site configuration to ensure that processes and procedures are kept the same between sites, making operational tasks easier to maintain and execute. Identical site configuration also allows for a higher success rate for manually maintaining the synchronization of Oracle Application Server component configuration files between sites.
- When the production site becomes unavailable due to a disaster, the standby site can become operational within a reasonable time. Client requests are always routed to the site that is operating in the production role. After a failover or switchover operation occurs due to an outage, client requests are routed to another site that assumes the production role. For a symmetric topology, the quality of service offered by the new production site should be the same as that offered by the original production site before the outage.
- The sites are set up in active-passive configuration. An active-passive setup has one primary site used for production and one secondary site that is initially passive (on standby). The secondary site is made active only after a failover or switchover operation is performed. Since the sites are symmetrical, after failover or switchover, the original standby site can be kept active as the new production site. After repairing or upgrading the original production site, it can be made into the new standby site as long as the OracleAS Disaster Recovery site requirements are maintained. Either site should offer the same level of service to clients as the other.
- The site playing the standby role contains a physical standby of the Oracle Application Server Infrastructure coordinated by Oracle Data Guard, OracleAS Guard automates the configuration and use of Oracle Data Guard together with procedures for backing up and restoring OracleAS Infrastructure configuration files and provides configuration synchronization between the production and standby sites. Switchover and failover operations allow the roles to be traded between the OracleAS Infrastructures in the two sites. Refer to Section 11.8, "OracleAS Guard Operations -- Standby Site Cloning of One or More Production Instances to a Standby System", Section 11.9, "OracleAS Guard Operations -- Standby Instantiation and Standby Synchronization", Section 11.10, "Runtime Operations -- OracleAS Guard Switchover and Failover Operations", and Section 11.13, "Using OracleAS Guard Command-Line Utility (asgctl)" for information about using the asgctl command-line interface to perform OracleAS Guard administrative tasks of cloning, instantiation, synchronization, switchover, and failover in the OracleAS Disaster Recovery solution.

11.1.3.2 Asymmetrical Topologies - Simple Asymmetric Standby Topology with Collocated Oracle Identity Management and OracleAS Metadata Repository Infrastructure

Beginning with OracleAS Disaster Recovery Release 10.1.2.0.2, support for asymmetric topologies includes support for the following simple asymmetric standby topologies:

- A standby site having reduced resources (fewer middle tiers); this means support for all production services except the scaling. This approach guarantees all services

are maintained, but not scaled (see Figure 11–2 for an example of this OracleAS Disaster Recovery solution).

Figure 11–2 Simple Asymmetric Standby with Reduced Resources

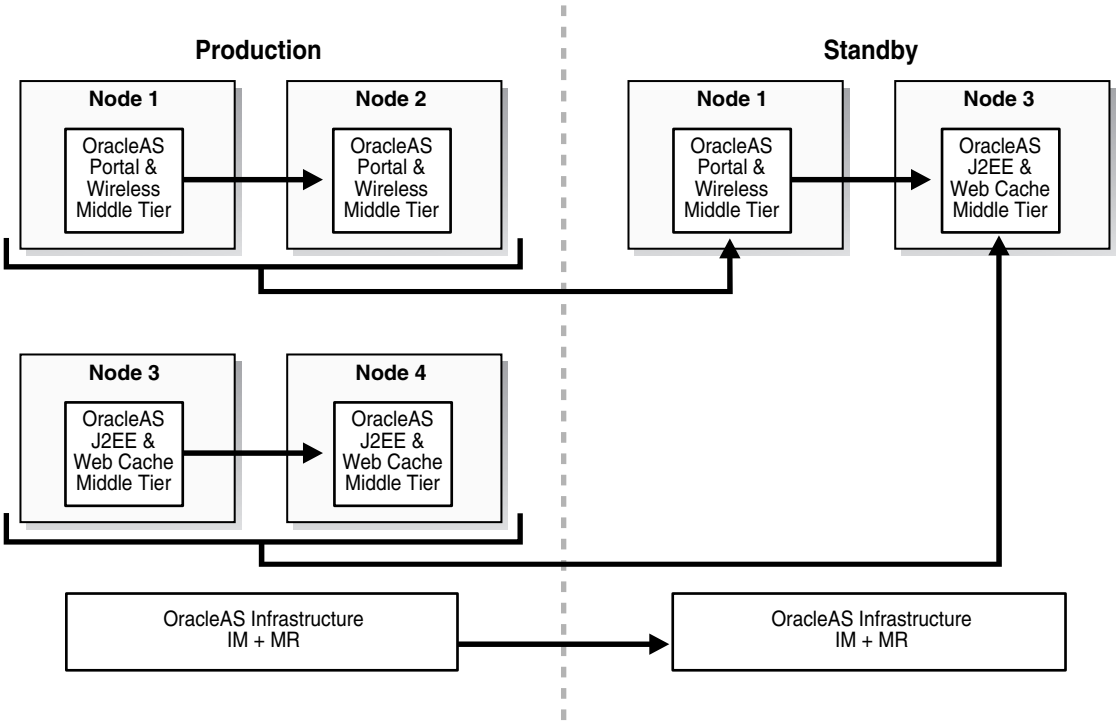


Figure 11–2 shows a production site of four middle tier instances and one Infrastructure (collocated Oracle Identity Management and OracleAS Metadata Repository). In this example, the services and applications deployed to middle tier 1 are scaled to include middle tier 2. In addition, the services and applications deployed to middle tier 3 are scaled to include middle tier 4. To satisfy the requirements for reduced resources for disaster recovery, the scaling is not necessary at the standby site. Therefore, the services deployed at production middle tiers 1 and 2 are satisfied by a disaster recovery peer middle tier 1 at the standby site, which will be synchronized with the production middle tier 1. Likewise, the services deployed at production middle tiers 3 and 4 are satisfied by a disaster recovery peer middle tier 3 at the standby site, which will be synchronized with the production middle tier 3.

- A standby site that maintains OracleAS Disaster Recovery support for the Infrastructure services only, while the middle-tier instances are supported only through production site management. This approach guarantees that only the Infrastructure services are maintained (see Figure 11–3 for an example of this OracleAS Disaster Recovery solution).

Figure 11–3 Simple Asymmetric Standby with Guaranteed Infrastructure

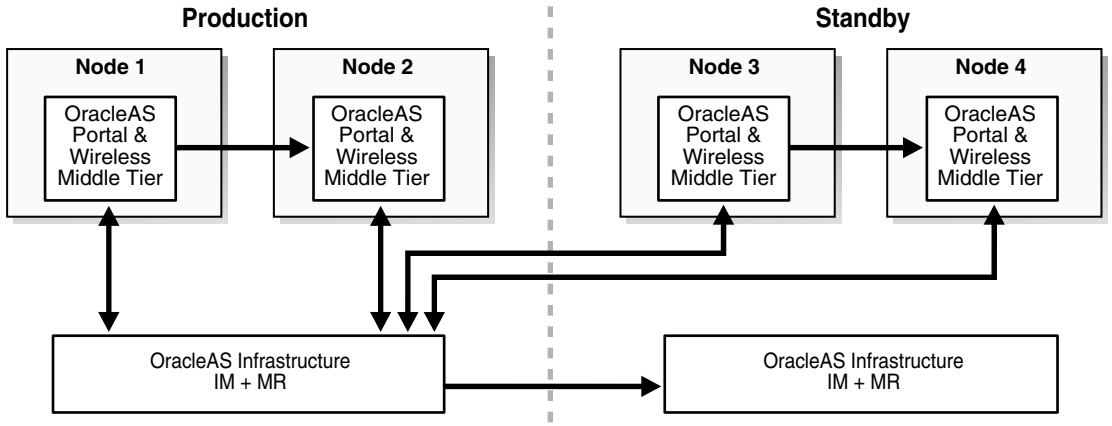


Figure 11–3 shows a production site consisting of four middle tiers instances, with two middle tiers (1 and 2) collocated with the production Infrastructure services and two middle tiers (3 and 4) remotely located at the standby site. The standby site is used to provide disaster recovery capability for only the Infrastructure services. In this configuration, middle tier resources are configured in an active/active model and technically as a single production topology.

Under normal conditions, application requests can be serviced from middle tiers 1 through 4. This model assumes that the services and applications deployed to middle tiers 3 and 4 can tolerate the latency, firewall, and network issues associated with this topology. For disaster recovery operations, only the Infrastructure services must be maintained, while the deployment and maintenance of the middle tier instances is done through routine production site management.

In general, support for asymmetrical topologies means that the OracleAS Disaster Recovery standby site has or potentially has reduced resources, maintains a reduction of Oracle homes, and also guarantees a certain minimum level of service capability.

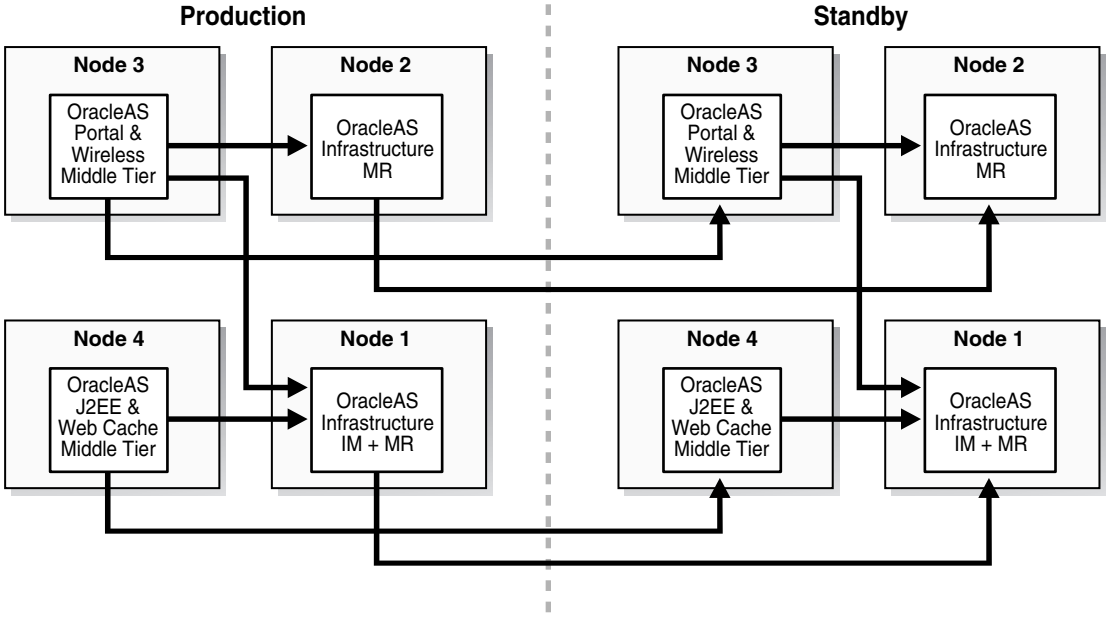
11.1.3.3 Separate OracleAS Metadata Repository for OracleAS Portal with Collocated Oracle Identity Management and OracleAS Metadata Repository Infrastructure (the Departmental Topology)

This topology (Figure 11–4), consists of an OracleAS Infrastructure with two OracleAS Metadata Repositories and multiple middle tiers. One OracleAS Metadata Repository is used by Oracle Identity Management components, such as Oracle Internet Directory and OracleAS Single Sign-On. All middle tiers use this OracleAS Metadata Repository for Oracle Identity Management services, as well as any additional middle tiers that might be added to this topology as it expands. The other OracleAS Metadata Repository is used for product metadata by the OracleAS Portal and OracleAS Wireless middle tier components. With two metadata repositories, this deployment can best be described as two DCM production farms.

An OracleAS Disaster Recovery standby configuration could be set up as either a symmetrical topology as described in Section 11.1.3.1, "Symmetrical Topologies - Strict Mirror of the Production Site with Collocated Oracle Identity Management and OracleAS Metadata Repository Infrastructure", thereby requiring two DCM standby farms be configured or as a simple asymmetric topology as described in Section 11.1.3.2, "Asymmetrical Topologies - Simple Asymmetric Standby Topology with Collocated Oracle Identity Management and OracleAS Metadata Repository

Infrastructure", with service guaranteed requiring minimally that a single DCM standby farm be configured.

Figure 11–4 Collocated Oracle Identity Management and OracleAS Metadata Repository with a Separate OracleAS Metadata Repository



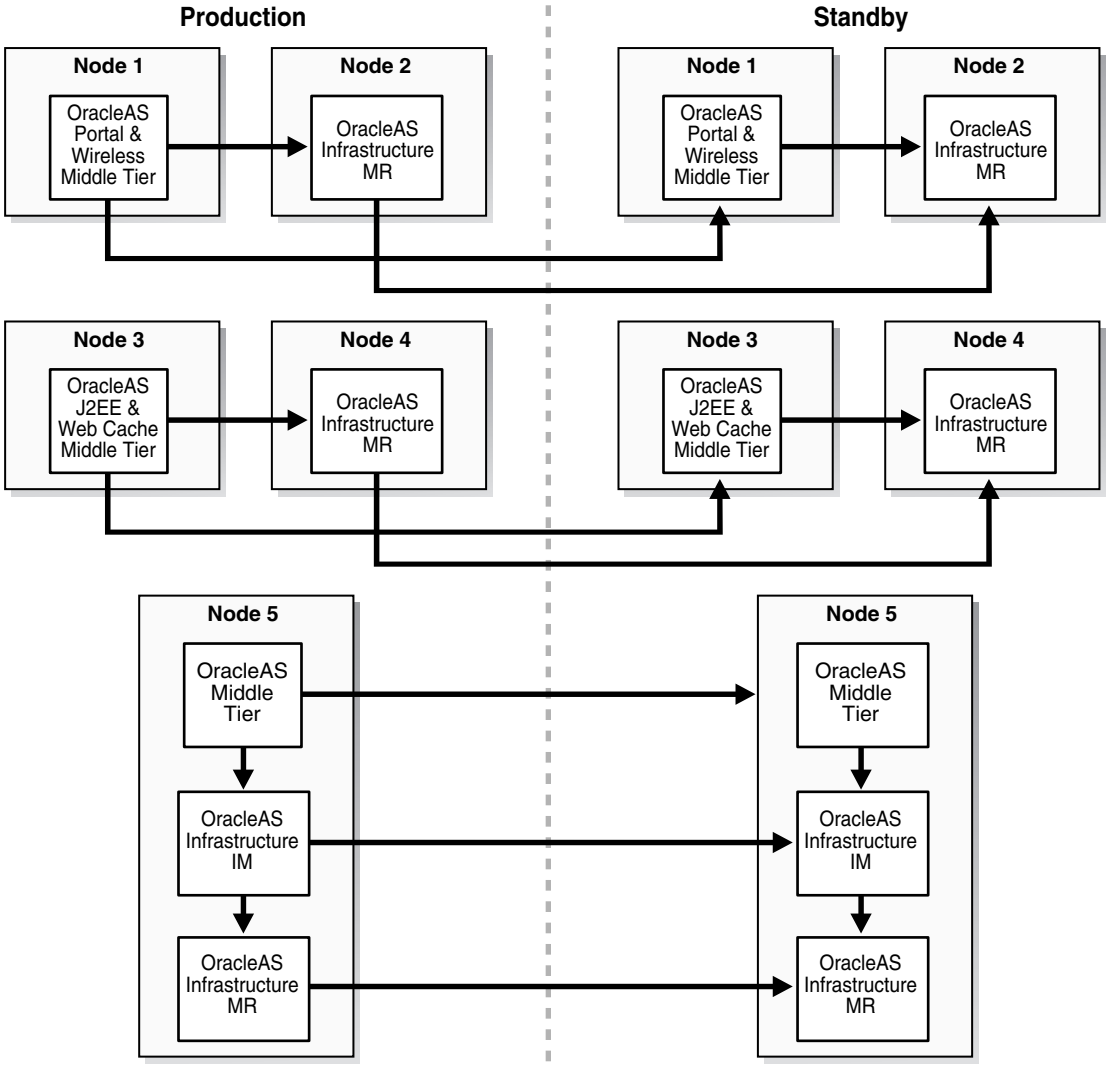
11.1.3.4 Distributed Application OracleAS Metadata Repositories with Non Collocated Oracle Identity Management and OracleAS Metadata Repository Infrastructure

The topologies in Section 11.1.3.1, "Symmetrical Topologies - Strict Mirror of the Production Site with Collocated Oracle Identity Management and OracleAS Metadata Repository Infrastructure", Section 11.1.3.2, "Asymmetrical Topologies - Simple Asymmetric Standby Topology with Collocated Oracle Identity Management and OracleAS Metadata Repository Infrastructure", and Section 11.1.3.3, "Separate OracleAS Metadata Repository for OracleAS Portal with Collocated Oracle Identity Management and OracleAS Metadata Repository Infrastructure (the Departmental Topology)" describe a deployment for a default database repository collocated for both the Oracle Identity Management and OracleAS Metadata Repository Infrastructure, while Section 11.1.3.3, "Separate OracleAS Metadata Repository for OracleAS Portal with Collocated Oracle Identity Management and OracleAS Metadata Repository Infrastructure (the Departmental Topology)" also describes a topology with a separate OracleAS Metadata Repository.

In a topology with distributed application OracleAS Metadata Repositories and non collocated Oracle Identity Management and OracleAS Metadata Repository Infrastructure, the Oracle Identity Management Infrastructure and one OracleAS Metadata Repository Infrastructure are installed on separate hosts, and other OracleAS Metadata Repositories are installed to reside with respective applications on different hosts. Thus, one OracleAS Metadata Repository can be the result of a deployment using a single default Infrastructure install, while one or more OracleAS Metadata Repositories can be the result of an OracleAS user using a tool, such as the OracleAS Metadata Repository Creation Assistant, to install one or more application OracleAS Metadata Repositories on one or more systems with the application data, for management or policy reasons, or both.

Figure 11–5 shows an example OracleAS Disaster Recovery solution having non collocated Oracle Identity Management and OracleAS Metadata Repository Infrastructure and distributed OracleAS Metadata Repositories.

Figure 11–5 Non-Collocated Oracle Identity Management (IM) and OracleAS Metadata Repository (MR) Infrastructure Topology with Distributed OracleAS Metadata Repositories



11.2 Preparing the OracleAS Disaster Recovery Environment

Prior to the installation of OracleAS software for the OracleAS Disaster Recovery solution, a number of system level configurations are required or optional as specified. The tasks that accomplish these configurations are:

- Section 11.2.1, "Planning and Assigning Hostnames"
- Section 11.2.2, "Configuring Hostname Resolution"
- Chapter 16, "Secure Shell (SSH) Port Forwarding" (optional)

This section covers the steps needed to perform these tasks for the symmetrical topology. These same steps are also applicable to simple asymmetrical standby sites as

well as to topologies for non collocated Oracle Identity Management and OracleAS Metadata Repository with or without distributed OracleAS Metadata Repositories.

11.2.1 Planning and Assigning Hostnames

Before performing the steps to set up the physical and network hostnames, plan the physical and network hostnames you wish to use with respect to the entire OracleAS Disaster Recovery solution. The overall approach to planning and assigning hostnames must meet the following goals:

- OracleAS components in the middle tier and OracleAS Infrastructure must use the same physical hostnames in their configuration settings regardless of whether the components are in the production or standby site. In addition, you must also create a virtual hostname for the physical hostname of the OracleAS Infrastructure.

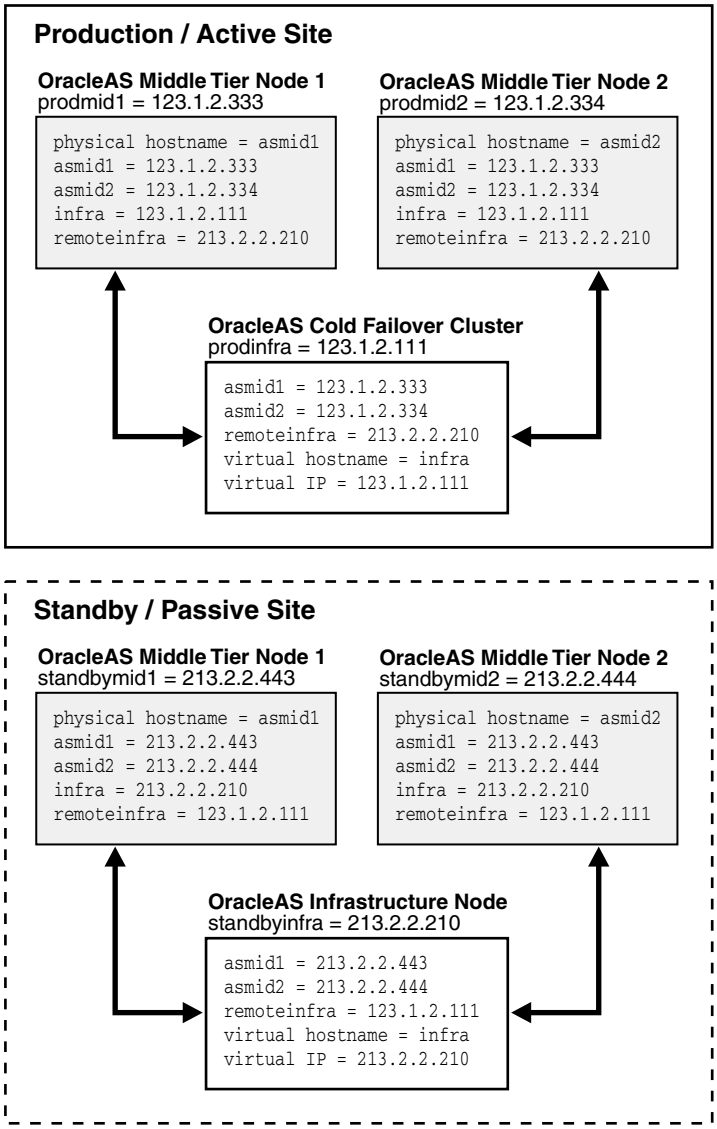
  For example, if a middle-tier component in the production site uses the name "`asmid1`" to reach a host in the same site, the same component in the standby site must use the same name to reach `asmid1`'s equivalent peer in the standby site. Likewise, if the virtual hostname of the OracleAS Infrastructure on the production site uses the name "`infra`", the virtual hostname for the physical hostname of the OracleAS Infrastructure on the standby site must be named "`infra`".

- No changes to hostnames (physical, network, or virtual) are required when the standby site takes over the production role. However, a DNS switchover must be performed, see Section 11.12, "Wide Area DNS Operations" for more information.

Note: Although the physical hostnames in the production and standby sites must remain uniform between the two sites, the resolution of these physical hostnames to the correct hosts can be different. Section 11.2.2, "Configuring Hostname Resolution" explains hostname resolution.

Figure 11–6 illustrates the process of planning and assigning hostnames.

Figure 11–6 Name Assignment Example in the Production and Standby Sites



In Figure 11–6, two middle-tier nodes exist in the production site. The OracleAS Infrastructure can be a single node or an OracleAS Cold Failover Cluster solution (represented by a single virtual hostname and a virtual IP, as for a single node OracleAS Infrastructure). The common names in the two sites are the physical hostnames of the middle-tier nodes and the virtual hostname of the OracleAS Infrastructure. Table 11–2 details what the physical, network, and virtual hostnames are in the example:

Table 11–2 Physical, network, and virtual hostnames in Figure 11–6

Physical Hostnames	Virtual Hostname	Network Hostnames
asmid1	-	prodmid1, standbymid1
asmid2	-	prodmid2, standbymid2
- [1]	infra	prodinfra, standbyinfra

[1] In this example, the physical hostname is the network hostname. Therefore, the network host name is used in the appropriate asgctl commands for the respective <host>, <host-name>, or <standby_topology_host> parameter arguments.

- *Cohosting non OracleAS applications*

  If the hosts in the production site are running non OracleAS applications, and you wish to cohost OracleAS on the same hosts, changing the physical hostnames of these hosts may break these applications. In such a case, you can keep these hostnames in the production site and modify the physical hostnames in the standby site to match those in the production site. The non OracleAS applications can then also be installed on the standby hosts so that they can act in a standby role for these applications.

As explained in Section 1.2.1, "Terminology", physical, network, and virtual hostnames have different purposes in the OracleAS Disaster Recovery solution. They are also set up differently. The following sections provide information about how the three types of hostnames are set up.

11.2.1.1 Physical Hostnames

The naming of middle-tier hosts in both the production and standby sites requires changing the physical hostname in each host.

In Solaris, to change the physical hostname of a host:

> **Note:** For other UNIX variants, consult your system administrator for equivalent commands in each step.

1. Check the setting for the existing hostname as follows:

   ```
   prompt> hostname
   ```

2. Use a text editor, such as `vi`, to edit the name in `/etc/nodename` to your planned physical hostname.
3. For each middle-tier host, reboot it for the change to take effect.
4. Repeat Step 1 to verify the correct hostname has been set.
5. Repeat the previous steps for each host in the production and standby sites.

In Windows, to change the physical hostname of a host, follow these steps:

> **Note:** The user interface elements in your version of Windows may vary from those described in the following steps.

1. In the Start menu, select **Control Panel**.
2. Double-click the System icon.
3. Select the **Advance** tab.
4. Select Environment variables.
5. Under the User Environment variables for the installer account, select **New** to create a new variable.
6. Enter the name of the variable as "`_CLUSTER_NETWORK_NAME_`".
7. For the value of this variable, enter the planned physical hostname.

11.2.1.2 Network Hostnames

The network hostnames used in the OracleAS Disaster Recovery solution are defined in domain name system (DNS). These hostnames are visible in the network that the solution uses and are resolved through DNS to the appropriate hosts by the assigned IP address in the DNS system. You need to add these network hostnames and their corresponding IP addresses to the DNS system.

Using the example in Figure 11–6, the following additions should be made to the DNS system serving the entire network that encompasses the production and standby sites:

```
prodmid1.oracle.com         IN     A     123.1.2.333
prodmid2.oracle.com         IN     A     123.1.2.334
prodinfra.oracle.com        IN     A     123.1.2.111
standbymid1.oracle.com      IN     A     213.2.2.443
standbymid2.oracle.com      IN     A     213.2.2.444
standbyinfra.oracle.com     IN     A     213.2.2.210
```

11.2.1.3 Virtual Hostname

As defined in Section 1.2.1, "Terminology", virtual hostname applies to the OracleAS Infrastructure only. It is specified during installation of the OracleAS Infrastructure. When you run the OracleAS Infrastructure installation type, a screen called "Specify High Availability" appears to provide a text box to enter the virtual hostname of the OracleAS Infrastructure that is being installed. Refer to the *Oracle Application Server Installation Guide* for more details.

For the example in Figure 11–6, when you install the production site's OracleAS Infrastructure, enter its virtual hostname, "`infra`", when you see the **Specify Virtual Hostname** screen. Enter the same virtual hostname when you install the standby site's OracleAS Infrastructure.

> **Note:** If the OracleAS Infrastructure is installed in an OracleAS Cold Failover Cluster solution, the virtual hostname is the name that is associated with the virtual IP of the OracleAS Cold Failover Cluster.

11.2.2 Configuring Hostname Resolution

In the OracleAS Disaster Recovery solution, you can configure hostname resolution in one of two ways to resolve the hostnames you planned and assigned in Section 11.2.1, "Planning and Assigning Hostnames". These are:

- Section 11.2.2.1, "Using Local Hostnaming File Resolution"
- Section 11.2.2.2, "Using DNS Resolution"

In UNIX, the order of the method of name resolution can be specified using the "`hosts`" parameter in the file `/etc/nsswitch.conf`. The following is an example of the `hosts` entry:

```
hosts:      files dns nis
```

In the previous statement, local hostnaming file resolution is preferred over DNS and NIS (Network Information Service) resolutions. When a hostname is required to be resolved to an IP address, the `/etc/hosts` file (UNIX) or `C:\WINDOWS\system32\drivers\etc\hosts` file is consulted first. In the event that a hostname cannot be resolved using local hostnaming resolution, DNS is used. (NIS resolution is not used for the OracleAS Disaster Recovery solution.) Refer to your

UNIX system documentation to find out more about name resolution using the file `/etc/nsswitch.conf`.

11.2.2.1 Using Local Hostnaming File Resolution

This method of hostname resolution relies on a local hostnaming file to contain the requisite hostname-to-IP address mappings. In UNIX, this file is `/etc/hosts`. In Windows, this file is `C:\WINDOWS\system32\drivers\etc\hosts`.

To use the local hostnaming file to resolve hostnames for the OracleAS Disaster Recovery solution in UNIX for each middle tier and OracleAS Infrastructure host in both the production and standby sites, perform the following steps:

1. Use a text editor, such as `vi`, to edit the `/etc/nsswitch.conf` file. With the "`hosts:`" parameter, specify "`files`" as the first choice for hostname resolution.
2. Edit the `/etc/hosts` file to include the following:
   - The physical hostnames and the correct IP addresses for all middle-tier nodes in the current site. The first entry must be the hostname and IP address of the current node.

     Note: When making entries in the hosts file, make sure the intended hostname is positioned in the second column of the hosts file; otherwise, an asgctl `verify topology with <host>` operation will fail indicating that the production topology is not symmetrical with the standby topology. See Appendix A, "Troubleshooting High Availability" for more information about troubleshooting and resolving this type of problem.

     For example, if you are editing the `/etc/hosts` file of a middle-tier node in the production site, enter all the middle-tier physical hostnames and their IP addresses in the production site beginning the list with the current host. (You should also include fully qualified hostnames in addition to the abbreviated hostnames. See Table 11–3.)
   - The virtual hostname of the OracleAS Infrastructure in the current site.

     For example, if you are editing the `/etc/hosts` of a middle-tier node in the standby site, enter the virtual hostname, fully qualified and abbreviated, and the IP address of the OracleAS Infrastructure host in the standby site.
3. Reboot each host after editing the files mentioned in the previous steps.
4. From each host, use the ping command for each physical hostname that is valid in its particular site to ensure that the IP addresses have been assigned correctly.

   For the example in Figure 11–6, on the `asmid1` host, use the following commands in succession:

   ```
   ping asmid1
   ```

   The returned IP address should be 123.1.2.333.

   ```
   ping asmid2
   ```

   The returned IP address should be 123.1.2.334.

   ```
   ping infra
   ```

   The returned IP address should be 123.1.2.111.

Note: Some UNIX variants, such as Solaris, require the `-s` option to return an IP address.

In Windows, the method of ordering hostname resolution varies depending on the Windows version. Refer to the documentation for your version of Windows for the appropriate steps.

Using the example in Figure 11–6, Table 11–3 shows that the `/etc/hosts` file entries on each production node contains the required entries in the of each UNIX host. The entries in the Windows `C:\WINDOWS\system32\drivers\etc\hosts` file should be similar.

Table 11–3 Network and Virtual Hostname Entries in Each `/etc/hosts` File of Example in Figure 11–6

Host	Entries in /etc/hosts

`asmid1` in production site	`123.1.2.333 asmid1.oracle.com asmid1`
	`123.1.2.334 asmid2.oracle.com asmid2`
	`123.1.2.111 infra.oracle.com infra`
	`213.2.2.210 remoteinfra.oracle.com remoteinfra`

`asmid2` in production site	`123.1.2.334 asmid2.oracle.com asmid2`
	`123.1.2.333 asmid1.oracle.com asmid1`
	`123.1.2.111 infra.oracle.com infra`
	`213.2.2.210 remoteinfra.oracle.com remoteinfra`

`infra` in production site	`123.1.2.111 infra.oracle.com infra`
	`123.1.2.333 asmid1.oracle.com asmid1`
	`123.1.2.334 asmid2.oracle.com asmid2`
	`213.2.2.210 remoteinfra.oracle.com remoteinfra`

`asmid1` in standby site	`213.2.2.443 asmid1.oracle.com asmid1`
	`213.2.2.444 asmid2.oracle.com asmid2`
	`213.2.2.210 infra.oracle.com infra`
	`123.1.2.111 remoteinfra.oracle.com remoteinfra`

`asmid2` in standby site	`213.2.2.444 asmid2.oracle.com asmid2`
	`213.2.2.443 asmid1.oracle.com asmid1`
	`213.2.2.210 infra.oracle.com infra`
	`123.1.2.111 remoteinfra.oracle.com remoteinfra`

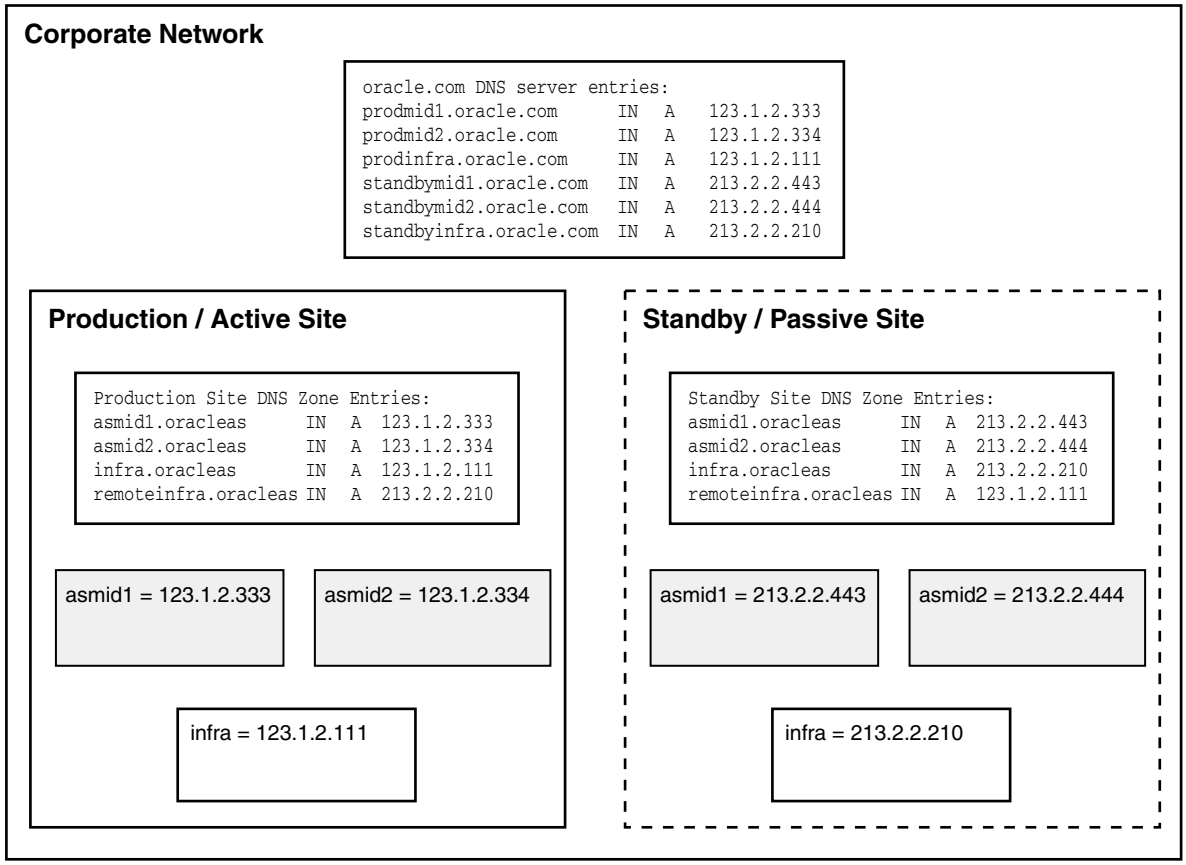
`infra` in standby site	`213.2.2.210 infra.oracle.com infra`
	`213.2.2.443 asmid1.oracle.com asmid1`
	`213.2.2.444 asmid2.oracle.com asmid2`
	`123.1.2.111 remoteinfra.oracle.com remoteinfra`

11.2.2.2 Using DNS Resolution

To set up the OracleAS Disaster Recovery solution to use DNS hostname resolution, you must set up site-specific DNS servers in the production and standby sites in addition to the overall corporate DNS servers (usually more than one DNS server exists in a corporate network for redundancy). Figure 11–7 provides an overview of this setup.

See Also: Chapter 15, "Setting Up a DNS Server" for instructions on how to set up a DNS server in a UNIX environment.

Figure 11–7 DNS Resolution Topology Overview



For the topology in Figure 11–7 to work, the following requirements and assumptions must be made:

- The DNS servers for the production and standby sites must not be aware of each other. They make non authoritative lookup requests to the overall corporate DNS servers if they fail to resolve any hostnames within their specific sites.
- The production site and standby site DNS servers must contain entries for middle-tier physical hostnames and OracleAS Infrastructure virtual hostnames. Each DNS server contains entries of only the hostnames within its own site. The sites have a common domain name that is different from that of the overall corporate domain name.
- The overall corporate DNS servers contain network hostname entries for the middle-tier hosts and OracleAS Infrastructure hosts of both production and standby sites.
- In UNIX, the `/etc/hosts` file in each host does not contain entries for the physical, network, or virtual hostnames of any host in either the production or standby site. In Windows, this applies to the file `C:\WINDOWS\system32\drivers\etc\hosts`.

To set up the OracleAS Disaster Recovery solution for DNS resolution, follow these steps:

1. Configure each of the overall corporate DNS servers with the network hostnames of all the hosts in the production and standby sites. Using the example presented in Figure 11–6, the following entries are made:

```
prodmid1.oracle.com      IN    A    123.1.2.333
prodmid2.oracle.com      IN    A    123.1.2.334
prodinfra.oracle.com     IN    A    123.1.2.111
standbymid1.oracle.com   IN    A    213.2.2.443
standbymid2.oracle.com   IN    A    213.2.2.444
standbyinfra.oracle.com  IN    A    213.2.2.210
```

2. For each site, production and standby, create a unique DNS zone by configuring a DNS server as follows:

   a. Select a unique domain name to use for the two sites that is different from the corporate domain name. As an example, use the name "`oracleas`" for the domain name for the two sites in Figure 11–6. The high level corporate domain name is `oracle.com`.

   b. Configure the DNS server in each site to point to the overall corporate DNS servers for unresolved requests.

   c. Populate the DNS servers in each site with the physical hostnames of each middle-tier host and the virtual hostname of each OracleAS Infrastructure host. Include the domain name selected in the previous step.

      For the example in Figure 11–6, the entries are as follows:

      For the DNS server on the production site:

      ```
      asmid1.oracleas      IN    A    123.1.2.333
      asmid2.oracleas      IN    A    123.1.2.334
      infra.oracleas       IN    A    123.1.2.111
      ```

      For the DNS server on the standby site:

      ```
      asmid1.oracleas      IN    A    213.2.2.443
      asmid2.oracleas      IN    A    213.2.2.444
      infra.oracleas       IN    A    213.2.2.210
      ```

Note: If you are using the OracleAS Cold Failover Cluster solution for the OracleAS Infrastructure in either site, enter the cluster's virtual hostname and virtual IP address. For example, in the previous step, `infra` is the virtual hostname and `123.1.2.111` is the virtual IP of the cluster in the production site. For more information on the OracleAS Cold Failover Cluster solution, see Section 4.2, "Common Characteristics of OracleAS Cold Failover Cluster Topologies" on page 4-12.

11.2.2.2.1 Additional DNS Server Entries for Oracle Data Guard

Because OracleAS Guard automates the use of Oracle Data Guard technology, which is used to synchronize the production and standby OracleAS Infrastructure databases, the production OracleAS Infrastructure must be able to reference the standby OracleAS Infrastructure and conversely.

For this to work, the IP address of the standby OracleAS Infrastructure host must be entered in the production site's DNS server with a hostname that is unique to the production site. Similarly, the IP address of the production OracleAS Infrastructure host must be entered in the standby site's DNS server with the same hostname. These DNS entries are required because Oracle Data Guard uses TNS Names to direct requests to the production and standby OracleAS Infrastructures. Hence, the appropriate entries must also be made to the `tnsnames.ora` file. Additionally,

OracleAS Guard asgctl command-line commands must reference the network hostnames.

Using the example in Figure 11–6 and assuming that the selected name for the remote OracleAS Infrastructure is "`remoteinfra`," the entries for the DNS server in the production site are:

```
asmid1.oracleas         IN    A    123.1.2.333
asmid2.oracleas         IN    A    123.1.2.334
infra.oracleas          IN    A    123.1.2.111
remoteinfra.oracleas    IN    A    213.2.2.210
```

And, in the standby site, the DNS server entries should be as follows:

```
asmid1.oracleas         IN    A    213.2.2.443
asmid2.oracleas         IN    A    213.2.2.444
infra.oracleas          IN    A    213.2.2.210
remoteinfra.oracleas    IN    A    123.1.2.111
```

11.3 Overview of Installing Oracle Application Server

This section provides an overview of the steps for installing the OracleAS Disaster Recovery solution. These steps are applicable to the topologies described in Section 11.1.3, "Supported Topologies". After following the instructions in Section 11.2, "Preparing the OracleAS Disaster Recovery Environment" to set up the environment for the solution, read this section for an overview of the installation process. Then, follow the detailed instructions in the *Oracle Application Server Installation Guide* to install the solution.

Note: To assign identical ports for use by symmetrical hosts in the production and standby sites, you can use static port definitions. These definitions are defined in a file, (for example, named `staticports.ini`). Then, specify the `staticports.ini` file in the **Specify Ports Configuration Options** screen in the installer. (Detailed information on the static ports file is found in the *Oracle Application Server Installation Guide*.)

The following steps represent the overall sequence for installing the OracleAS Disaster Recovery solution:

1. Install OracleAS Infrastructure in the production site (see *Oracle Application Server Installation Guide*).
2. Install OracleAS Infrastructure in the standby site (see *Oracle Application Server Installation Guide*).
3. Start the OracleAS Infrastructure in each site before installing the middle tiers for that site.
4. Install the middle tiers in the production site (see *Oracle Application Server Installation Guide*).
5. Install the middle tiers in the standby site (see *Oracle Application Server Installation Guide*).

The following points are important when you perform the installation:

- Ensure that the same ports are used by equivalent peer hosts in both sites. For example, the `asmid1` host in the standby site must use the same ports as the `asmid1` host in the production site. Use a static ports definition file. (see the previous note in this section and the following point).
- Specify the full path to the `staticports.ini` file in the installer's **Specify Ports Configuration Options** screen.
- Ensure that you select the High Availability and Replication option in the installer's **Select Configuration Options** screen.
- Specify the virtual address assigned to the OracleAS Infrastructure in the **Specify Virtual Hostname** screen during OracleAS Infrastructure installation.
- Install for the middle-tier hosts, any of the available middle-tier installation types. (Ensure that the OracleAS Infrastructure services have been started for a site before installing any middle tiers in that site.)
- Specify the OracleAS Infrastructure's virtual hostname as the OracleAS Infrastructure database during each middle-tier installation.
- Start the OracleAS services on the hosts in each site starting with the OracleAS Infrastructure.

11.4 Overview of OracleAS Guard and asgctl

This section provides an overview of OracleAS Guard and its command-line interface asgctl. If you are already familiar with this overview information, go to Section 11.5, "Authentication of Databases". This section contains the following subsections:

11.4.1 Overview of asgctl

The asgctl command-line utility greatly simplifies the complexity and magnitude of the steps involved in setting up and managing OracleAS Disaster Recovery. This utility provides a distributed solution that consists of a client component and a server component. The client component (OracleAS Guard client) can be installed on a system on the topology. The server component (OracleAS Guard server) is installed by default on the systems hosting the primary and standby Oracle homes that comprise the OracleAS Disaster Recovery environment.

11.4.2 OracleAS Guard Client

The OracleAS Guard client is installed on every OracleAS install type. The OracleAS Guard client attempts to open and maintain a connection to the OracleAS Guard server.

The OracleAS Guard client provides an asgctl command-line interface (CLI) (see Chapter 12, "OracleAS Guard asgctl Command-line Reference") consisting of a set of

commands to perform administrative tasks described in Section 11.4.4, "asgctl Operations".

11.4.3 OracleAS Guard Server

The OracleAS Guard server is a distributed server (installed by default) that runs on all the systems in an OracleAS Disaster Recovery configuration. The OracleAS Guard client maintains an active connection to the OracleAS Guard server on one system that has network connectivity in the OracleAS Disaster Recovery configuration. This coordinating server communicates to the OracleAS Guard servers on the other systems in the OracleAS Disaster Recovery configuration as necessary to complete processing during standby site cloning, instantiation, synchronization, verification, switchover, and failover operations. The OracleAS Guard server carries out asgctl commands issued directly by the OracleAS Guard client or issued on behalf of the OracleAS Guard client by another OracleAS Guard server in the network for the client session. The steps to complete an operation will execute throughout all systems in both the production and standby topologies. Most operational steps will be executed either in parallel or sequentially (as required) on these systems throughout the OracleAS Disaster Recovery configuration by the OracleAS Guard server.

11.4.4 asgctl Operations

Major asgctl operations using the asgctl commands belong in the following categories of operations:

- Authentication -- Identify the OracleAS Infrastructure database on the primary topology (set primary database command). If there are topologies with multiple Infrastructures, each must be identified using this command prior to performing an operation involving both production and standby topologies.

  Identify the new OracleAS Infrastructure database on the standby topology (set new primary database command) prior to a failover operation.

  Set the credentials (set asg credentials command) used to authenticate the OracleAS Guard client connections to OracleAS Guard servers and the connections between OracleAS Guard servers to a specific host. See the set asg credentials command for an example, and see Section 11.14.1.1, "Setting asgctl Credentials" for more information.

  When OracleAS Guard discovers the topology (discover topology command), it requires you provide Oracle Internet Directory authentication credentials (Oracle Internet Directory password) in order to query Oracle Internet Directory to obtain instance information for the production site.

- Discover the topology -- Discover (discover topology command) by querying Oracle Internet Directory all instances within the topology that share the same Oracle Internet Directory for a production site and generate a topology XML file that describes the topology and replicates this file to all instances in the topology. See Section 11.6, "Discovering, Dumping, and Verifying the Topology" for more information.

  The command discover topology within farm discovers the topology using OPMN at a production site for special cases where Oracle Internet Directory is not available.

- Standby site cloning -- Clone a single production instance to a standby system (clone instance command) or clone two or more production instances to standby systems (clone topology command). See Section 11.8, "OracleAS Guard Operations -- Standby Site Cloning of One or More Production Instances to a Standby System"

for more information. The standby site cloning operation eliminates the task of having to install these Oracle instances on the standby middle tier systems and perform an instantiate operation.

- Standby site instantiation -- Creates the disaster recovery environment. It establishes the relationship between standby and production instances, mirrors the configuration, create the standby Infrastructure, then synchronize the standby site with the primary site (instantiate topology command). See Section 11.9.1, "Standby Instantiation" for more information.
- Standby site synchronization -- Applies database redo logs for OracleAS Infrastructures to the standby site in conjunction with synchronizing external configuration files across the topology (sync topology command). See Section 11.9.2, "Standby Synchronization" for more information.
- Switchover -- Switch from the production site to the standby site after the standby site is synchronized with the production site with the application of the database redo logs (switchover topology command). See Section 11.10.1.1, "Scheduled Outages" for more information.
- Failover -- Make the standby site the production site after restoring configuration files and restoring the OracleAS server environment to the point of the last successful sync operation (failover command). See Section 11.10.1.2, "Unplanned Outages" for more information.
- Verification -- Validate that the primary topology is running and the configuration is valid (verify topology command) or if a standby topology is specified, compare the primary topology to which the local host system is a member with the standby topology to validate that they are consistent with one another and conform to the requirements for OracleAS Disaster Recovery. See Section 11.11.1, "Verifying the Topology" for more information.
- Using a policy file -- Used as a filter to filter out unnecessary instances for supporting asymmetric topologies. The dump policies command writes detailed, default policy information to respective XML formatted files for a select set of asgctl commands. You can then edit each respective XML policy file and use it in the `using policy <file>` parameter with any one of these select set of asgctl commands: dump topology, verify topology, clone topology, failover, instantiate topology, switchover topology, and sync topology to define by instance the domain of execution operations that are permitted for each of these asgctl commands. Each instance list entry in an XML policy file logically tags a production-standby peer combination with a particular attribute that defines the success requirement for its successful operation. For example, you may want to omit a node in a symmetric topology while performing one of the operations previously mentioned. Use the policy file to specify the node to be ignored. See Section 11.7, "Dumping Policy Files and Using Policy Files With Some asgctl Commands" for more information.
- Instance management -- Enables you to shut down (shutdown topology command) and start up the topology (startup topology command).
- Troubleshooting -- Uses the dump topology command to write detailed information about the topology to the screen or to a file. Lets you determine the current operations that are running (show operation command) and stop any operations that need to be halted (stop operation command).

Table 11–4 describes the OracleAS Disaster Recovery production and standby site environment before and after performing an asgctl clone, instantiate, sync, failover, and switchover operation.

Table 11–4 Description of Disaster Recovery Production and Standby Environments Before and After Performing These OracleAS Guard Operations

OracleAS Guard Operation	DR Site Environment Before Operation	DR Site Environment After Operation
clone	The production site has one or more instances that need to be installed on the standby site and instantiated. The cloning operations perform this task.	The standby site has one or more new standby instances that are a logical mirror of the production site instances.
instantiate	The standby site with its Oracle homes exists, but the OracleAS Disaster Recovery relationship across sites does not exist yet for an OracleAS Disaster Recovery operation to be performed.	A logical mirror of the production site is set up and maintained at the standby site.
sync	The standby site is not consistent with the production site. OracleAS Disaster Recovery is not possible to restore the standby site to a consistent point in time without some manual intervention.	Database redo logs are applied to OracleAS Infrastructures in combination with synchronizing external configuration files across the topology. The sync operation is performed in the event that a failover or switchover operation is necessary, then the standby site can be restored to a consistent point in time. No manual intervention is necessary to synchronize the sites after the asgctl sync operation is performed.
switchover	A planned outage at the production site will make the standby site the production site for a period of time; that is the roles of each site will be switched.	The standby site has become the production site. All OPMN services are started. The production site may become available again after the planned outage, at which time, another switchover operation could be performed to return activity back to the original production site from the standby site.
failover	An unscheduled outage at the production site has left the production site down or unavailable for an unknown period of time. The production site is lost due to some unforeseen circumstance.	The standby site has permanently become the production site. Configuration and Infrastructure data are restored to a consistent point in time on the standby site. Site services are brought up in a consistent fashion to the point of the last sync operation. All OPMN services are started.

11.4.5 OracleAS Guard Integration with OPMN

A typical Oracle Application Server site has multiple farms. OracleAS Guard server and its ias-component DSA process is not started by default by OPMN because it is only necessary in the context of disaster recovery sites. You must start this ias-component DSA process in all Oracle homes as described later in this section. To check the status of this component and determine if the component is running, run the following opmnctl command on each system in your topology:

```
On UNIX systems
> <ORACLE HOME>/opmn/bin/opmnctl status

On windows systems
> <ORACLE HOME>\opmn\bin\opmnctl status
```

Because there is no way an OracleAS Guard client nor OPMN on the production site can start OracleAS Guard services on the standby site, OracleAS Guard must be started directly using opmnctl on the Infrastructure node in the standby topology. Connect to a node and run the following OPMN command on UNIX systems:

```
> <ORACLE HOME>/opmn/bin/opmnctl startproc ias-component=DSA
```

On Windows systems, issue the following OPMN command to start OracleAS Guard if your Oracle home is located on drive C:.

```
C:\<ORACLE HOME>\opmn\bin\opmnctl startproc ias-component=DSA
```

After the OracleAS Guard server is started it is non transient, while the remaining OracleAS Guard servers in the standby topology are transient servers. This configuration allows cross-topology communication.

> **Note:** When you perform an opmnctl status command on a system on which OracleAS Guard is running, you will see an ias-component and process-type named DSA. This is the OracleAS component name and server process name for the OracleAS Guard server.

11.4.6 Supported OracleAS Disaster Recovery Configurations

For OracleAS 10g release (10.1.2), OracleAS Guard supports not only the default OracleAS Infrastructure configuration supported on Oracle Application Server Cold Failover Cluster and single instance, but also the topologies described in Section 11.1.3, "Supported Topologies".

11.4.7 Configuring OracleAS Guard and Other Relevant Information

By default, OracleAS Guard and asgctl, the command-line utility for OracleAS Guard, are installed for all install types with the following default configuration information, which includes:

- The following OracleAS Guard parameters are configurable. The value is described and the default value is indicated. The OracleAS Guard `readme.txt` file in the `<ORACLE_HOME>\dsa\doc` directory also lists these OracleAS Guard parameters that are configurable.
  - `port` - the TCP/IP port for OracleAS Guard server and client. OracleAS Guard uses a default port (port) number of 7890; for example, port=7890. If there is a second Oracle home installed on a system, this second Oracle home must have a different OracleAS Guard port number, usually incremented by one, for example, port=7891, and so on.

    Value: integer, any valid TCP/IP port number. Default is 7890.
  - `exec_timeout_secs` - timeout value for executing operating system command.

    Value: integer, number of seconds. Default is 60 seconds.
  - `trace_flags` - trace flags to be turned on.

    Value: string list, separated by ",". Default is none.
  - `backup_mode` - indicates whether to perform a full or incremental backup.

    Value: String, "full" or "incremental". Default is "incremental".
  - `backup_path` - the backup directory path to be used by OracleAS Guard server.

    Value: string, a directory path. Default is `<ORACLE_HOME>/dsa/backup`.
  - `ha_path` - the High Availability directory path where the backup scripts are located.

Value: string, a directory path. Default is `<ORACLE_HOME>/backup_restore`.

- `port.<host>` - the TCP/IP port for a given host.

  Value: integer, any valid TCP/IP port number.

  Note: If the port number must be changed for some reason (it must be unique for each OracleAS Guard server in each Oracle home on a machine, which is automatically handled during installation), you can change its value in the `<ORACLE_HOME>/dsa/dsa.conf` file. Then, stop the OracleAS Guard server(`<ORACLE_HOME>/opmn/bin/opmnctl stopproc ias-component=DSA`) and start the OracleAS Guard server (`<ORACLE_HOME>/opmn/bin/opmnctl startproc ias-component=DSA`) to activate the change. See Section 11.4.5, "OracleAS Guard Integration with OPMN") for more information.

- `copyfile_buffersize` - the buffer size for copy file operation, in kilobytes.

  Value: integer, maximum buffer size is 500K.

- `server_inactive_timeout` - the number of seconds server will wait before shutting down due to inactivity.

  Value: integer, number of seconds. Default value is 600 seconds (10 minutes).

- `inventory_location` - the alternative Oracle Inventory location

  Value: string, the full path of the location of `oraInst.loc` file.

- OracleAS Guard command-line utility asgctl is installed in the `<ORACLE_HOME>/dsa/bin` directory on UNIX systems and `<ORACLE_HOME>\dsa\bin` directory on Windows systems on all nodes in the topology production and standby topologies.
- OracleAS Guard starts up the OracleAS component services across the production topology.
- The OracleAS Guard operation status information for a topology (from either an asgctl show operation full or show operation history command) remains available for the life of the current OracleAS Guard client asgctl connect session only. When the OracleAS Guard client disconnects from the OracleAS Guard server, this topology's operation history information becomes unavailable.
- After you start an asgctl operation, you cannot run another asgctl command on the same OracleAS Guard server until the previous command that is running completes or is forced to stop (see the asgctl stop operation command for more information.) In addition, you cannot run an asgctl operation in background and then quit or exit the asgctl utility.

11.5 Authentication of Databases

Several levels of authentication are required when an OracleAS Guard client connects to an OracleAS guard server and begins a session to perform administrative operations within the production topology or across both production and standby topologies:

- Infrastructure authentication

- OracleAS Guard client authentication to OracleAS Guard servers
- Oracle Internet Directory authentication

Infrastructure Authentication

When initiating an OracleAS Guard administrative session, after establishing the connection between the OracleAS Guard client and OracleAS Guard server, you must identify all the OracleAS Infrastructure databases on the primary topology using the set primary database command. Infrastructure authentication must be performed before you initiate any operation that involves either the production topology or both the production and standby topologies.

Another form of Infrastructure authentication occurs as part of a failover operation. In this scenario, the production site is down and you must failover to the standby site and make this site the new production site. First, identify the new OracleAS Infrastructure database on the standby topology using the set new primary database command before performing the failover operation. See Section 11.10.1.2, "Unplanned Outages" for more information.

OracleAs Guard Client Authentication to OracleAS Guard Servers

By default, these are the same authentication credentials used for instance level authentication with the Oracle Application Server account (`ias_admin`/password) that was created during the Oracle Application Server installation and used in the connect asg command. These same credentials are used when the OracleAS Guard client connects to any OracleAS Guard server in the production and standby topology when executing administrative operations.

There may be cases where you want to use different credentials for a specific OracleAS Guard server or set a common set of credentials in the standby topology that differs from the credentials used in the primary topology. To set credentials for an OracleAS Guard server, use the set asg credentials command and one or more of its parameter options by either specifying the host name to which the credentials apply or the topology along with the new set of credentials (username/password).

If you set the credentials for a topology, these credentials are inherited for the entire topology. If you set the credentials for an individual host on the topology, the credentials for this host override the default credentials set for the topology. After you set the credentials so that they are different from the default connection credentials for a host system or an entire topology, whenever you initiate an OracleAS Guard administrative session, you must specify all credentials that are different from the default connection credentials for any host system or topology before you perform an operation involving all the OracleAS Guard servers within a production topology or across both production and standby topologies. Otherwise, the operation will fail with an authentication error. See the connect asg command for an example.

Oracle Internet Directory Authentication

The discover topology command requires you provide Oracle Internet Directory authentication credentials (Oracle Internet Directory password) in order to query Oracle Internet Directory to obtain instance information for the production site. See the section that follows for more information and the discover topology command.

11.6 Discovering, Dumping, and Verifying the Topology

The discover topology command discovers by querying Oracle Internet Directory all instances within the topology that share the same Oracle Internet Directory for a production site. A topology XML file is created and distributed to all Oracle homes

within the topology that describes all instances for the topology. This topology file is used by all OracleAS Guard operations.

You must perform a discover topology command when you first set up your OracleAS Disaster Recovery environment in order to initially create the topology XML file. Thereafter, you should perform a discover topology operation whenever you procure another Oracle home in a production site or change roles from a production to a standby site through a switchover or failover operation. See the discover topology command for more information.

You should perform a dump topology command to inspect the information that describes your topology. See the dump topology command for more information.

You should perform a verify topology command to validate that the primary topology is running and that the configuration is valid. In addition, if you specify the `with host` parameter, the verify operation compares the primary topology of which the local host system is a member with the standby topology to validate that they are consistent with one another and conform to the requirements for OracleAS Disaster Recovery. See Section 11.11.1, "Verifying the Topology" and the verify topology command for more information.

With both the dump topology and verify topology commands, if you want to use a policy file, edit and use the respective dump and verify policy files (`dump_policy.xml` and `verify_policy.xml`). Specify this file in the `using policy <file>` parameter of each command to dump or verify only those instances specified accordingly. See Section 11.7, "Dumping Policy Files and Using Policy Files With Some asgctl Commands" for more information.

11.7 Dumping Policy Files and Using Policy Files With Some asgctl Commands

OracleAS Disaster Recovery provides support for a variety of application server topologies as described in Section 11.1.3, "Supported Topologies". As part of this support, a set of XML formatted policy files are maintained, local to the OracleAS Guard client that performs the dump policies command, to record by instance the domain of execution operations that are permitted for each of the following asgctl commands: dump topology, verify topology, clone topology, failover, instantiate topology, switchover topology, and sync topology.

To understand the default policies in use for any these asgctl commands, enter the following command at the asgctl prompt:

```
ASGCTL> dump policies
Generating default policy for this operation
Creating policy files on local host in directory
"/private1/OraHome2/asr1012/dsa/conf/"
ASGCTL>
```

Each instance list entry in each of the XML policy files logically tags by default a production-standby peer combination with a particular attribute that defines the success requirement for the successful operation of each command. This approach provides greater flexibility in regulating how each of these OracleAS Guard operations are to be successfully used among the supported topologies, see Section 11.1.3, "Supported Topologies" for more information.

After inspecting each of the XML formatted policy files, you can subsequently edit the respective policy file and use it with the particular asgctl command using the parameter syntax `using policy <file>` and indicate the name of the policy file to

be used. In this way, you can employ a particular disaster recovery policy that defines the success requirement attribute value by instance for each of these OracleAS Guard operations mentioned earlier in this chapter.

> **Note:** If you want to maintain a set of custom policy files, you must copy, edit, and maintain them in a location other than the default location; otherwise, your custom set of policy files will be overwritten whenever you perform a discover topology command followed subsequently by a dump policies command.

The success requirement attribute value can be one of the following: [`optional | mandatory | ignore | group <MinSucceeded=<number>>`], where:

- `Optional` -- means if there is a failure for that instance continue processing other instances.
- `Mandatory` -- means if an error occurs for this instance, the entire operation fails.
- `Ignore` -- means the instance is not part of the operation.
- `Group <MinSucceeded=<number>` -- means to combine groups of Oracle instances, and if the specified number of group members is successful, then the operation is successful; otherwise, if less than the number of group members that is specified is successful, the operation fails.

Each attribute value determines the success requirement for that peer group and will be referenced during failure cases of asgctl operations to determine whether or not to continue with the OracleAS Guard operation. For example, when the success requirement is specified as mandatory, the particular OracleAS Guard operation must be successful for the specified instance for that production-standby peer combination; otherwise, the OracleAS Guard operation ceases, execution is rolled back to its starting point of execution, and an error message is returned.

For example, the following XML policy file in use for an asymmetric topology for the failover operation specifies that this asgctl operation is mandatory for the infra instance, optional for the portal_1 and portal_2 instances, can be ignored for the portal_3 instance, and must be successful for a minimum of any two of the group of three instances, BI_1, BI_2, and BI_3.

```
<policy>
  <instanceList successRequirement="Mandatory">
     <instance>infra</instance>
  </instanceList >
  <instanceList successRequirement="Optional">
     <instance>portal_1</instance>
     <instance>portal_2</instance>
  </instanceList >
  <instanceList successRequirement="Ignore">
     <instance>portal_3</instance>
  </instanceList >
  <instanceList successRequirement="Group" minSucceed="2">
     <instance>BI_1</instance>
     <instance>BI_2</instance>
     <instance>BI_3</instance>
  </instanceList >
</policy>
```

11.8 OracleAS Guard Operations -- Standby Site Cloning of One or More Production Instances to a Standby System

Standby site cloning is the process of cloning a single production instance to a standby system (using the clone instance command) or cloning two or more production instances to standby systems (using the clone topology command).

Clone Instance

The clone instance command is used to create a new standby instance target from an existing production instance source.

One of the underlying technologies used by OracleAS Guard to perform this operation is the OracleAS Backup and Restore loss of host capability. See the section on recovering a loss of host automatically in *Oracle Application Server Administrator's Guide* for more information including a list of prerequisites. This capability assumes that the target machine is a newly procured Oracle environment because it overwrites the Oracle software registry. Additionally, some of the underlying operations require elevated privileges, `root` for the UNIX environments and `Administrator` for Windows. On Windows, the user must ensure that the client and OracleAS Guard server are started with `Administrator` privileges.

There are two phases of clone. The first phase is to create the Oracle home and register it within the system environment. The second phase is to perform the OracleAS Guard instantiate operation to link it into the OracleAS Disaster Recovery environment and logically match the Oracle home with it's corresponding production home.

A series of clone instance operations on different instances are equivalent to a clone topology operation.

Clone Topology

The clone topology command performs a clone instance operation across a group of systems. The clone operation is performed on every OracleAS home that does not contain a database or it can be filtered using a policy file. For OracleAS homes that contain a database, a clone topology operation will perform the instantiate phase of the operation, skipping the creation of the Oracle home at the standby site. The operation can be performed on a subset of a topology by utilizing a policy file.

There are three methodologies that you must be aware of when planning for an OracleAS Disaster Recovery site setup:

- Creating a pure OracleAS Disaster Recovery site
- Adding OracleAS homes to an existing site with OracleAS Disaster Recovery enabled
- Integrating OracleAS Metadata Repositories within an existing database

Each operation requires a different methodology to integrate the newly installed Oracle homes into the existing site or combine them into a standby site for a production site.

Creating a Pure OracleAS Disaster Recovery Site

Prior to OracleAS 10g release 10.1.2.0.2, this was the only type of site OracleAS Guard could support. An OracleAS Disaster Recovery configuration was supported only for the default Infrastructure and OracleAS middle-tier install types. With this type of configuration, all the OracleAS homes were created using the Oracle installer. The OracleAS Guard instantiate command creates the relationships between the

production and standby Oracle homes and the underlying standby Oracle database repositories.

Adding OracleAS Homes to an Existing Site with OracleAS Disaster Recovery Enabled

After an OracleAS site is OracleAS Disaster Recovery enabled, the relationship between the production and standby Oracle homes has been created. For releases previous to OracleAS 10g release 10.1.2.0.2, the only way to add new instances to the site was to break the standby relationship, add the new instance at the production site using Oracle Installer, add the new instance to the standby site using Oracle Installer, and re-create the standby site. With OracleAS 10g release 10.1.2.0.2, you can use the clone instance command to add instances to a standby site.

For example, if you need a new middle tier to scale out the services in the middle tier the new instance is installed at the production site. This operation creates the OracleAS home for the instance and establishes the necessary relationships within the OracleAS repositories.

With OracleAS 10g release 10.1.2.0.2, OracleAS Guard asymmetrical topology support, this Oracle home can optionally be ignored in regard to the site's OracleAS Disaster Recovery solution. If you want to add this instance to the standby site, the clone topology command will create the OracleAS Oracle home at the standby target host and establish the production-standby relationship for this instance. Before issuing this command, the standalone OracleAS Guard kit must be installed and started at the target host (see the OracleAS Disaster Recovery installation information in *Oracle Application Server Installation Guide* for more information) and a site discovery topology operation should be performed to discover the new instance in the production topology.

Integrating OracleAS Metadata Repositories within an Existing Database

OracleAS supports the ability to create Metadata Repository schemas in an existing database. Although OracleAS Guard recognizes and manages these databases to synchronize the Metadata Repository configuration data with the rest of the site's distributed configuration data, OracleAS Guard does not create the standby repository nor the production to standby relationship. This environment is supported using the clone topology operation.

To utilize the clone topology command, first install and start the standalone OracleAS Guard server on each standby host. Additionally, the OracleAS Backup/Restore utility is installed in the Oracle home created by the standalone OracleAS Guard install. See the section on recovering a loss of host automatically in *Oracle Application Server Administrator's Guide* for more information including a list of prerequisites. The clone topology command creates the middle-tier instance Oracle homes and configuration information at the standby site. For Infrastructure instances, an implicit instantiate operation is performed to initialize the OracleAS Disaster Recovery environment. It is assumed that a separate OracleAS install has already been performed on the standby host. The clone topology operation can use a profile file to filter out instances for an asymmetric topology.

Warning: Do not perform a clone operation to a standby system that contains an existing Oracle home, other than the standalone OracleAS Guard home, because it will get overwritten. Perform a clone operation only to a standby system where no Oracle home is installed other than the standalone OracleAS Guard home.

Some situations in which cloning operations are useful are:

- When you want to add one or more production instances to a standby host site.
- When you want to add a single production instance to a standby host system.

The steps to perform these cloning operations are described in the following sections.

11.8.1 Cloning a Single Production Instance to a Standby System

As an example, you want to add a production instance to a standby system. The clone instance operation eliminates the task of having to install the Oracle instance on the standby middle-tier system and then perform an instantiate operation.

The production instance to be cloned cannot exist on the standby system.

The following are prerequisites for performing the clone instance operation to the standby site system:

- The OracleAS Guard standalone kit must be installed on the standby system.
- Backup and Restore must be installed in the OracleAS Guard home on the standby system.
- A Java development kit with its jar utility must be installed on the standby system.
- For Windows systems, the services kit (`sc.exe`) must be installed on the standby system.

The basic procedure consists of the following pre-clone, clone, and post-clone steps.

Pre-Clone Steps

For each instance on the production and standby sites, perform the following steps:

1. Log in as su - root on UNIX systems or as Administrator on Windows systems.
2. CD to the instance home.
3. Shut down any OracleAS Guard servers.

```
On UNIX systems:
> <ORACLE HOME>/opmn/bin/opmnctl stopproc ias-component=DSA

On Windows systems:
C:\<ORACLE HOME>\opmn\bin\opmnctl stopproc ias-component=DSA
```

4. **On UNIX systems only:** make sure `dsaServer.sh` in *<ORACLE_HOME>*`/dsa/bin` is executable by everyone. If it is not, record the permission, then change the executable permission by issuing the following command:

```
chmod +x dsaServer.sh
chmod u+x asgexec
```

5. Invoke asgctl and issue the startup command.

```
>On UNIX systems from the <ORACLE_HOME>/dsa/bin directory
> asgctl.sh startup

On Windows systems fom the <ORACLE_HOME>\dsa\bin directory
C:\> asgctl startup
```

6. Log out as root on UNIX systems.

Clone Steps

From any instance on the production site, perform the following steps:

1. Log in as user (non root user on UNIX systems).
2. CD to the production instance home.
3. Invoke asgctl and run the clone instance command to clone the instance to the standby topology host system.

```
> asgctl.sh
Application Server Guard: Release 10.1.2.0.2
(c) Copyright 2004, 2005 Oracle Corporation. All rights reserved
ASGCTL> connect asg prodinfra ias_admin/adminpwd
Successfully connected to prodinfra:7890
ASGCTL> set primary database sys/testpwd@asdb
Checking connection to database asdb
ASGCTL> clone instance portal_2 to asmid2
Generating default policy for this operation
.
.
.
ASGCTL> disconnect
ASGCTL> exit
>
```

4. Log out of the system.

Post-Clone Steps

For the instance on the production and standby sites, perform the following steps:

1. Log in as su - root on UNIX systems or as Administrator on Windows systems.
2. CD to the instance home.
   - On the production site systems, CD to the instance home.
   - On the standby site systems, CD to the OracleAS Guard standalone home.
3. Perform an asgctl shutdown command.

```
>On UNIX systems from the <ORACLE_HOME>/dsa/bin directory
> asgctl.sh shutdown

On Windows systems fom the <ORACLE_HOME>\dsa\bin directory
C:\> asgctl shutdown
```

4. Log out as root on UNIX systems.
5. **On UNIX systems only:** Restore the permission for `dsaServer.sh` to what you recorded it as in Pre-Clone Step 4.
6. On the standby site only, CD to the newly cloned home.
7. Start up OracleAS Guard using the following opmnctl command:

```
On Unix systems:
> <ORACLE HOME>/opmn/bin/opmnctl startproc ias-component=DSA

On Windows systems:
C:\<ORACLE HOME>\opmn\bin\opmnctl startproc ias-component=DSA
```

Note: If OracleAS Guard does not run as root on UNIX systems, the user will be prompted by the OracleAS Guard client to run the underlying operations at each of the instance homes as root (manually) in order to continue with the operation.

The last step completes the cloning instance operation and brings the systems back to where they were before you started the operation. At this point, you could invoke asgctl, connect to a production system, discover the topology, and then perform a verify operation to determine if the production and standby topologies were valid and consistent with one another as you would expect them to be.

11.8.2 Cloning Multiple Production Instances to Standby Systems

As an example, you want to add two or more production instances to a standby middle-tier host system. The clone topology operation eliminates the task of having to install these Oracle instances on the standby middle-tier systems and then perform an instantiate operation.

As part of the clone topology operation, the production instances are cloned and the OracleAS Metadata Repository is instantiated. However, for a OracleAS Metadata Repository configuration created using OracleAS Metadata Repository Creation Assistant, no instantiate operation is performed.

The production instances to be cloned cannot exist on the standby systems.

If you want to use a policy file, edit and use the clone policy file (`clone_policy.xml`). Specify this file in the `using policy <file>` parameter of the clone topology command to clone a standby topology for only those instances specified accordingly. See Section 11.7, "Dumping Policy Files and Using Policy Files With Some asgctl Commands" for more information.

The following are prerequisites for performing the clone topology operation to standby site systems:

- The OracleAS Guard standalone kit must be installed on each standby system.
- Backup and Restore must be installed on each OracleAS Guard home on each standby system.
- A Java development kit with its jar utility must be installed on each standby system.
- For Windows systems only, the services kit (`sc.exe`) must be installed on each standby system.

The basic procedure consists of the following pre-clone, clone, and post-clone steps.

Pre-Clone Steps

For each instance on the production and standby sites, perform the following steps:

1. Log in as su - root on UNIX systems or as Administrator on Windows systems.
2. CD to the instance home.
3. Shut down any OracleAS Guard servers.

```
On UNIX systems:
> <ORACLE HOME>/opmn/bin/opmnctl stopproc ias-component=DSA

On Windows systems:
```

```
C:\<ORACLE HOME>\opmn\bin\opmnctl stopproc ias-component=DSA
```

4. **On UNIX systems only:** make sure `dsaServer.sh` in *<ORACLE_HOME>*`/dsa/bin` is executable by everyone. If it is not, record the permission, then change the executable permission by issuing the following command:

```
chmod +x dsaServer.sh
chmod u+x asgexec
```

5. Invoke asgctl and issue the startup command.

```
>On UNIX systems from the <ORACLE_HOME>/dsa/bin directory
> asgctl.sh startup

On Windows systems fom the <ORACLE_HOME>\dsa\bin directory
C:\> asgctl startup
```

6. Log out as root on UNIX systems.

Clone Steps

From any instance on the production site, perform the following steps:

1. Log in as user (non root user on UNIX systems).
2. CD to any production instance home.
3. Invoke asgctl and run the clone topology command to clone the topology to the standby topology host system.

```
> asgctl.sh
Application Server Guard: Release 10.1.2.0.2
(c) Copyright 2004, 2005 Oracle Corporation. All rights reserved
ASGCTL> connect asg prodinfra ias_admin/adminpwd
Successfully connected to prodinfra:7890
ASGCTL> set primary database sys/testpwd@asdb
Checking connection to database asdb
ASGCTL>
# Command to use if you are using a policy file where <file>
# is the full path and file spec of the clone policy file.
ASGCTL> clone topology to standbyinfra using policy <file>
Generating default policy for this operation
.
.
.
ASGCTL> disconnect
ASGCTL> exit
>
```

4. Log out from the system.

Post-Clone Steps

For each instance on the production and standby sites, perform the following steps:

1. Log in as su - root on UNIX systems or as Administrator on Windows systems.
2. CD to the instance home.
   - On the production site systems, CD to the instance home.
   - On the standby site systems, CD to the OracleAS Guard standalone home.
3. Perform an asgctl shutdown command.

```
>On UNIX systems from the <ORACLE_HOME>/dsa/bin directory
> asgctl.sh shutdown

On Windows systems fom the <ORACLE_HOME>\dsa\bin directory
C:\> asgctl shutdown
```

4. Log out as root on UNIX systems.
5. **On UNIX systems only:** Restore the permission for `dsaServer.sh` to what you recorded it as in Pre-Clone Step 4.
6. On the standby site only, CD to the newly cloned homes.
7. Start up OracleAS Guard using the following opmnctl command:

```
On Unix systems:
> <ORACLE HOME>/opmn/bin/opmnctl startproc ias-component=DSA

On Windows systems:
C:\<ORACLE HOME>\opmn\bin\opmnctl startproc ias-component=DSA
```

> **Note:** If OracleAS Guard does not run as root on UNIX systems, the user will be prompted by the OracleAS Guard client to run the underlying operations at each of the instance homes as root (manually) in order to continue with the operation.

The last step completes the cloning topology operation and brings the systems back to where they were before you started the operation. At this point, you could invoke asgctl, connect to a production system, discover the topology, and then perform a verify operation to determine if the production and standby topologies were valid and consistent with one another, as you would expect them to be.

11.8.3 Cloning When There Are Multiple Instances on One System

When you are cloning a topology and there are two or more instances on a production system, multiple DSA ports are configured, with one DSA port uniquely configured for each production instance in each respective `dsa.conf` file; however, there is only one configured DSA port on the standby site configured for the single standby instance. How does OracleAS Guard resolve this problem?

To answer this question, lets consider the following example. Assume that on the production site `host1` there are two production instances, `instance_1` using DSA port 7890 and `instance_2` using DSA port 7891. Then, lets assume that on the standby site `host2`, the OracleAS Guard standalone kit is installed there and is using DSA port 7890.

By default, `instance_2` on production site `host1` will try to connect to the standby site `host2` using DSA port 7891. Because there is no standby OracleAS Guard server on the standby site using DSA port 7891, the `dsa.conf` file on production site `host1` for `instance_2` needs an entry in its `dsa.conf` file to resolve to DSA port 7890 before performing the clone operation as follows:

```
port.host2 = 7890
```

Making this entry in the `instance_2` `dsa.conf` file must precede the first pre-clone step (see Pre-Clone Steps in Section 11.8.2).

Then, after the clone operation completes, immediately following step 2 (see Post-Clone Steps in Section 11.8.2), this edited entry must be removed from the `dsa.conf` file for `instance_2` and the OracleAS Guard server stopped (see Step 3) and restarted (see Step 7) on production site `host1` for `instance_2`.

11.9 OracleAS Guard Operations -- Standby Instantiation and Standby Synchronization

After adhering to the following conditions, you are ready to use the Oracle Application Server Guard for standby instantiation and standby synchronization.

- Meet the requirements for the implementation of the OracleAS Disaster Recovery solution as described in Section 11.1.1, "OracleAS Disaster Recovery Requirements", Section 11.1.3, "Supported Topologies", and Section 11.2, "Preparing the OracleAS Disaster Recovery Environment".
- Install the OracleAS Disaster Recovery (DR) solution as described in Section 11.3, "Overview of Installing Oracle Application Server".

The following subsections describe standby instantiation and standby synchronization.

See Chapter 12, "OracleAS Guard asgctl Command-line Reference" for OracleAS Guard command-line asgctl utility reference information.

11.9.1 Standby Instantiation

The standby instantiation operation performs a number of operations to set up and maintain a logical mirror of the production site at the standby site. OracleAS Guard is used to coordinate the distributed operations across the production and standby sites to ensure the disaster recovery functionality is enabled. The setup operations are:

- Uses a previous topology file created by performing a discovery topology operation.
- Verifies the topology definitions to ensure they comply with the rules of the OracleAS Disaster Recovery environment.
- Configures Oracle Data Guard to maintain the OracleAS Disaster Recovery environment for the database repository.
- Mirrors the configuration information of all the Oracle homes in the OracleAS topology to the corresponding Oracle home at the standby site.
- If you want to use a policy file, edit and use the instantiate policy file (`instantiate_policy.xml`). Specify this file in the `using policy <file>` parameter of the instantiate topology command to instantiate a standby topology for only those instances specified accordingly. See Section 11.7, "Dumping Policy Files and Using Policy Files With Some asgctl Commands" for more information.
- Reports any errors found for correction.

The procedure to perform a standby instantiation operation uses the following example, which assumes that you have invoked the OracleAS Guard client and performed a discover topology command to create a topology file.

See Section 12.2.1.1, "Special Considerations for Running Instantiate and Failover Operations in CFC Environments" if you have an OracleAS Disaster Recovery configuration in a CFC environment and are about to perform an instantiate operation.

1. Connect to the OracleAS Guard server.

```
ASGCTL > connect asg prodinfra ias_admin/<adminpwd>
Successfully connected to prodinfra:7890
ASGCTL>
```

2. Specify the primary OracleAS Metadata Repository database. See Section 11.13.1.2, "Specifying the Primary Database" for more information. If you have multiple OracleAS Metadata Repositories in your topology, you must authenticate each one using the set primary database command.

   ```
   ASGCTL> set primary database sys/testpwd@asdb
   ```

3. Dump the policies (dump policies command), then edit and use the verify policy file (`verify_policy.xml`) and the instantiate policy file (`instantiate_policy.xml`) to specify the success requirement attribute for each instance in the file. See Section 11.7, "Dumping Policy Files and Using Policy Files With Some asgctl Commands" for more information.

   ```
   ASGCTL> dump policies
   Generating default policy for this operation
   Creating policy files on local host in directory
   "/private1/OraHome2/asr1012/dsa/conf/"
   ```

4. Verify the topology. The network hostname `standbyinfra` is used.

   ```
   ASGCTL> verify topology with standbyinfra
   ```

5. Instantiate the topology at the secondary site. The network hostname `standbyinfra` is used. This command assumes that all Oracle homes have been installed using Oracle installer software. Specify the `using policy <file>` parameter where `<file>` represents the path and file specification for the `instantiate_policy.xml` file.

   ```
   ASGCTL> instantiate topology to standbyinfra using policy <file>
   ```

Whenever a standby instantiation is performed using the asgctl instantiate topology command a synchronization operation is also performed. Thus, you do not need to perform another synchronization operation immediately following the instantiation operation. If a period of time had passed following an instantiate operation, ensure that both the primary and standby sites are consistent. Then, perform a sync topology operation to ensure any changes that occurred on the primary site are applied to the secondary site.

11.9.2 Standby Synchronization

The OracleAS Guard synchronization operation synchronizes the standby site with the primary site to ensure that the two sites are logically consistent. This operation is necessary whenever any of the following circumstances exist:

- Deploy a new application or redeploy an existing application - Both the deployment of a new application and the redeployment of an existing application require changes to schema-based information in the metadata repository as well as component configuration information distributed among the Oracle homes in an OracleAS topology. This information has to be uniformly deployed at the standby site.
- Configuration changes - Specific changes, small to large, to a configuration, must be reflected at the standby site.
- User Provisioning - The default Infrastructure installation maintains the database for Oracle Internet Directory. As new users are added to the database, they should

be synchronized to the standby site on a schedule that fulfills the business availability requirements.

- Periodic full synchronization - By default, the synchronization operations synchronizes only the pieces of configuration that have changed since the last synchronization operation. During test cycles or occasional complex configuration changes, administrators may want to fully refresh of their configuration information to the standby site to ensure mirroring of these changes.

You can specify a full or incremental synchronization. By default, an incremental synchronization is performed, which offers the best performance. However, in the following three circumstances a full synchronization should be specified:

- When you want to force a full synchronization to happen for some reason, such as synchronizing the standby site completely with the primary site.
- When you know there are many transactional changes over a short period of time on the primary site that must be synchronized with the secondary site.
- When you know that there is a large accumulation of transactional changes over a long period of time on the primary site that must be synchronized with the secondary site.

As part of the synchronization operation, a verify operation is performed to ensure the required OracleAS Disaster Recovery environment is maintained. Additionally, if new OracleAS instances are installed into the OracleAS topology, OracleAS Guard will discover these installations.

If you want to use a policy file, edit and use the synchronization policy file (`sync_policy.xml`). Specify this file in the `using policy <file>` parameter of the sync topology command for synchronizing a standby topology for only those instances specified accordingly. See Section 11.7, "Dumping Policy Files and Using Policy Files With Some asgctl Commands" for more information.

The following example assumes that you have invoked the OracleAS Guard client and performed a discover topology command to create a topology file.

The procedure to perform standby synchronization is as follows:

1. Connect to the OracleAS Guard server.

```
ASGCTL > connect asg prodinfra ias_admin/<adminpwd>
Successfully connected to prodinfra:7890
ASGCTL>
```

2. Specify the primary database. See Section 11.13.1.2, "Specifying the Primary Database" for more information.

```
ASGCTL> set primary database sys/testpwd@asdb
```

3. Synchronize the secondary site with the primary site.

```
ASGCTL> sync topology to standbyinfra
```

11.10 Runtime Operations -- OracleAS Guard Switchover and Failover Operations

Runtime operations include dealing with outages, whether they are scheduled or unscheduled (see Section 11.10.1, "Outages"), and monitoring ongoing OracleAS Guard operations using the asgctl command-line utility and troubleshooting (see Section 11.11, "Monitoring OracleAS Guard Operations and Troubleshooting").

11.10.1 Outages

Outages fall into two categories scheduled and unplanned.

The following subsections describe these outages.

11.10.1.1 Scheduled Outages

Scheduled outages are planned outages. They are required for regular maintenance of the technology infrastructure supporting the business applications and include tasks such as hardware maintenance, repair and upgrades, software upgrades and patching, application changes and patching, and changes to improve the performance and manageability of systems. Scheduled outages can occur either for the production or standby site. Descriptions of scheduled outages that impact the production or standby site are:

- Site-wide maintenance

  The entire site where the current production resides is unavailable. Examples of site-wide maintenance are scheduled power outages, site maintenance, and regularly planned switchover operations.

- OracleAS Cold Failover Cluster cluster-wide maintenance

  This is scheduled downtime of the OracleAS Cold Failover Cluster for hardware maintenance. The scope of this downtime is the whole hardware cluster. Examples of cluster-wide maintenance are repair of the cluster interconnect and upgrade of the cluster management software.

- Testing and validating the standby site as a means to test OracleAS Disaster Recovery readiness.

For scheduled outages, a site switchover operation has to be performed, which is explained in the section that follows.

Site Switchover Operations

A site switchover is performed for planned outages of the production site. Both the production and standby sites have to be available during the switchover. The application of the database redo logs is synchronized to match the backup and restoration of the configuration files for the middle tier and OracleAS Infrastructure installations.

> **Note:** During a switchover operation, the `opmn.xml` file is copied from the primary site to the standby site. For this reason, the value of the TMP variable must be defined the same in the `opmn.xml` file on both the primary and standby sites, otherwise this switchover operation will fail with a message that it could not find a directory. Therefore, make sure the TMP variable is defined identically and resolves to the same directory structure on both sites before attempting a switchover operation.

During site switchover, considerations must be made to avoid long periods of cached DNS information. Modifications to the site's DNS information, specifically time-to-live (TTL), must be performed. See Section 11.12.2, "Manually Changing DNS Names" for instructions.

If you want to use a policy file, edit and use the switchover policy file (`switchover_policy.xml`). Specify this file in the `using policy <file>` parameter of the switchover topology command for switching over to the standby topology only those

instances specified accordingly. See Section 11.7, "Dumping Policy Files and Using Policy Files With Some asgctl Commands" for more information. This example does not show the use of a policy file.

See Section 12.2.1.3, "Special Considerations for Running a Switchover Operations in CFC Environments" if you have an OracleAS Disaster Recovery configuration in a CFC environment and are planning a switchover operation.

To switchover from the production site to the standby site, perform the following steps:

1. Reduce the wide area DNS TTL value for the site. See Section 11.12.2, "Manually Changing DNS Names" for more information.
2. On the primary Infrastructure system, make sure the emagent process is stopped. Otherwise, the following error may occur when doing a switchover operation because the emagent has a connection to the database:

   ```
   prodinfra: -->ASG_DGA-13051: Error performing a physical standby switchover.
   prodinfra: -->ASG_DGA-13052: The primary database is not in the proper state to
   perform a switchover.  State is "SESSIONS ACTIVE"
   ```

   On UNIX systems, stop the Application Server Control (iasconsole) and stop the emagent process, as follows:

   ```
   > <ORACLE_HOME>/bin/emctl stop iasconsole
   ```

   On UNIX systems, to check to see if there is an emagent process running, enter the following command:

   ```
   > ps -ef | grep emagent
   ```

   On UNIX systems, if after performing the stop iasconsole operation, the emagent process is still running, obtain the process ID (PID) as shown in the previous ps command, and stop the emagent process as follows:

   ```
   > kill -9 <emagent-pid>
   ```

   On Windows systems, open the Services control panel. Locate the OracleAS10gASControl service and stop this service.
3. Invoke the OracleAS Guard client command-line utility asgctl (on UNIX systems, `asgctl.sh` is located in *<ORACLE_HOME>*`/dsa/bin` and on Windows systems, `asgctl.bat` is located in *<ORACLE_HOME>*`\dsa\bin`.) and connect to the OracleAS Guard server.

   ```
   > asgctl.sh
   Application Server Guard: Release 10.1.2.0.2
   (c) Copyright 2004, 2005 Oracle Corporation. All rights reserved
   ASGCTL> connect asg prodinfra ias_admin/<adminpwd>
   ```
4. Specify the primary database. See Section 11.13.1.2, "Specifying the Primary Database" for more information.

   ```
   ASGCTL> set primary database sys/testpwd@asdb
   ```
5. For Oracle RAC Disaster Recovery deployments, shut down all instances prior to switchover.

> **Note:** This example creates a script to install in Oracle homes and utilizes the ASG distributed ASG scripting capabilities. This allows the system administrator to perform all switchover operations from within the asgctl utility. The srvctl utility stops all instances within the cluster.

a. Create the `shutdown_asdb_instance.sh` script and copy the script to the location indicated in the script. The location must be in the Oracle home.

```
#shutdown_asdb_instance.sh for asdb instance
#in /private/oracle/product/10.1.0/asdb on db_site1_node1 & db_site2_node1
export ORACLE_HOME=/private/oracle/product/10.1.0/asdb
$ORACLE_HOME/bin/srvctl stop instance -d asdb -i asdb2
```

b. Use the asgctl run command to run the shut down instance script.

```
ASGCTL> run at instance asdb shutdown_asdb_instance.sh
```

6. Switchover the topology to the secondary site. If you want to use a policy file, specify the `using policy <file>` parameter where `<file>` represents the path and file specification for the `switchover_policy.xml` file.

```
ASGCTL> switchover topology to standbyinfra
```

> **Note:** As part of the OracleAS Guard switchover operation, an implicit sync topology operation is performed to make sure the topologies are identical. In addition, all OPMN services are stopped and then restarted on the production site.

7. Disconnect from the *old* primary site OracleAS Guard server.

```
ASGCTL> disconnect
ASGCTL>
```

8. Perform a wide area DNS switchover to direct requests to the new production site based on one of the options presented in Section 11.12, "Wide Area DNS Operations".
9. Adjust the wide area DNS TTL to an appropriate value.

Special Switchover Operation Considerations

This section describes the following special considerations relating to the switchover operation.

- When performing a switchover operation from a primary site with two Oracle Identity Management instances running to a standby site representing an asymmetric topology with only one Oracle Identity Management instance running, which means that the other node is to be ignored on the switchover site, the system administrator must not only edit the `switchover_policy.xml` policy file to indicate that this other node is to be set to Ignore, but must also shutdown all processes running on that node in order for the switchover operation to be successful. For example, if the two Oracle Identity Management instances running on the primary site are im.machineA.us.oracle.com and im.machineB.us.oracle.com, and the other node (im.machineB.us.oracle.com) is to be ignored on the switchover site, the system administrator must also shutdown

all processes running on that node (im.machineB.us.oracle.com) in order for the switchover operation to succeed.

- When the discover topology command is issued following a switchover operation and the asymmetric standby site topology originally had one or more fewer middle tiers (for example, instA and instB) than there were in the original production site topology (instA, instB, and instC), a warning error message displays for each missing instance of a middle tier (instC, in this case). This warning error message is expected and can be ignored. When a discover topology command is issued following a switchover operation, OracleAS Server Guard reads the Oracle Internet Directory information, which is an exact copy of the original primary site Oracle Internet Directory information on this new primary site (former standby site). Because this Oracle Internet Directory information is identical to the original primary site Oracle Internet Directory information, when OracleAS Server Guard visits the host or home of each instance of these middle tiers to verify their existence, it discovers that some of the middle tiers do not exist, and issues warnings.

11.10.1.2 Unplanned Outages

An unplanned outage that impacts a production site occurs when it becomes unavailable and there is no possibility of restoring the production site to service within a reasonable period of time. This includes site-wide outages at the production site such as fire, flood, earthquake, or power outages.

Unplanned outages warrant performing a failover operation of the production site to the standby site.

Site Failover Operations

A site failover operation is performed for unplanned outages for the production site. Failover operations require the restoration of the configuration and Infrastructure data to a consistent point in time. OracleAS Guard ensures that the site services are brought up in a consistent fashion to the point of the last sync operation. A failover operation restores to the last synchronization point.

If you want to use a policy file, edit and use the failover policy file (`failover_policy.xml`). Specify this file in the `using policy <file>` parameter of the failover command for failing over to the standby topology only those instances specified accordingly. See Section 11.7, "Dumping Policy Files and Using Policy Files With Some asgctl Commands" for more information.

See Section 12.2.1.1, "Special Considerations for Running Instantiate and Failover Operations in CFC Environments" if you have an OracleAS Disaster Recovery configuration in a CFC environment and are about to perform a failover operation.

To fail over the production site to the standby site, follow these steps:

1. Connect to the OracleAS Guard server on the standby site. The network name is standbyinfra.

```
ASGCTL> connect asg standbyinfra ias_admin/<adminpwd>
Successfully connected to stanfbyinfra:7890
```

2. For Oracle RAC Disaster Recovery deployments, start up the databases.

Note: This example creates a script to install in Oracle homes and utilizes the ASG distributed ASG scripting capabilities. This allows the system administrator to perform all failover operations from within the asgctl utility. The srvctl utility starts all instances within the cluster.

---

a. Create the `start_asdb_db.sh` script and copy the script to the location indicated in the script. The location must be in the Oracle home.

```
#start_asdb_db.sh for asdb database
#in /private/oracle/product/10.1.0/asdb on db_site1_node1 & db_site2_node1
export ORACLE_HOME=/private/oracle/product/10.1.0/asdb
$ORACLE_HOME/bin/srvctl start database -d asdb
```

b. Use the asgctl run command to run the start database script.

```
ASGCTL> run at instance asdb start_asdb_db.sh
```

3. Specify that the primary OracleAS Metadata Repository database on the standby site is now identified as the *new* primary database on this *new* production site. The keyword **new** is shown as bold text in the following example to indicate its importance as a key word. If you have multiple OracleAS Metadata Repositories in your topology, you must authenticate each one using the set new primary database command.

```
ASGCTL> set new primary database sys/testpwd@asdb
```

4. Perform an asgctl failover operation.

```
ASGCTL> failover
```

5. Discover the topology. You must perform this operation to create a new topology file for this production site.

```
ASGCTL> discover topology oidpassword=oidpwd
```

11.11 Monitoring OracleAS Guard Operations and Troubleshooting

After setting up your OracleAS Disaster Recovery solution, and instantiating the standby topology, and synchronizing the standby topology, you can use the OracleAS Guard client command-line utility asgctl to issue commands through the coordinating OracleAS Guard server to monitor asgctl operations and perform troubleshooting tasks. A typical OracleAS Guard monitoring or troubleshooting session may involve the following tasks:

1. Section 11.11.1, "Verifying the Topology"
2. Section 11.11.2, "Displaying the Current Operation"
3. Section 11.11.3, "Displaying a List of Completed Operations"
4. Section 11.11.4, "Stopping an Operation"
5. Section 11.11.5, "Tracing Tasks"
6. Section 11.11.6, "Writing Information About the Topology to a File"

As asgctl commands are issued through the OracleAS Guard client and requests are then made to the coordinating OracleAS Guard server, the coordinating OracleAS

Guard server communicates these requests to the other OracleAS Guard servers in the production and standby topologies, and status messages are returned to the OracleAS Guard client as well as any error messages should a particular task encounter a problem. Section 11.11.7, "Error Messages" describes where you can obtain more information about these error messages.

11.11.1 Verifying the Topology

To validate that the primary topology is running and the configuration is valid, enter the following asgctl command at the asgctl prompt.

```
ASGCTL> connect asg ias_admin/iastest2
Successfully connected to prodinfra:7890
ASGCTL> discover topology oidpassword=oidpwd
ASGCTL> verify topology
Generating default policy for this operation
prodinfra:7890
     HA directory exists for instance asr1012.infra.us.oracle.com
asmid2:7890
     HA directory exists for instance asmid2.asmid2.us.oracle.com
asmid1:7890
     HA directory exists for instance asmid1.asmid1.us.oracle.com
ASGCTL>
```

If you want to use a policy file, edit and use the verify policy file (`verify_policy.xml`) to specify the success requirement attribute for each instance in the file. Then specify the `using policy <file>` parameter in the verify command where `<file>` represents the path and file specification for the `verify_policy.xml` file. See Section 11.7, "Dumping Policy Files and Using Policy Files With Some asgctl Commands" for more information.

To compare a primary topology to which the local host is a member with a standby topology and ensure that they are consistent with one another and that both topologies conform to OracleAS Disaster Recovery requirements, enter the following asgctl command at the asgctl prompt and specify the name of the standby host system.

```
ASGCTL> dump policies
Generating default policy for this operation
Creating policy files on local host in directory
"/private1/OraHome2/asr1012/dsa/conf/"

ASGCTL> verify topology with standbyinfra
Generating default policy for this operation
prodinfra:7890
     HA directory exists for instance asr1012.infra.us.oracle.com
asmid2:7890
     HA directory exists for instance asmid2.asmid2.us.oracle.com
asmid1:7890
     HA directory exists for instance asmid1.asmid1.us.oracle.com
standbyinfra:7890
     HA directory exists for instance asr1012.infra.us.oracle.com
asmid2:7890
     HA directory exists for instance asmid2.asmid2.us.oracle.com
asmid1:7890
     HA directory exists for instance asmid1.asmid1.us.oracle.com
prodinfra:7890
    Verifying that the topology is symmetrical in both primary and standby configuration
ASGCTL>

# Command to use if you want to use a policy file
# verify topology with standbyinfra using policy <file>
```

11.11.2 Displaying the Current Operation

To display the status of all the current operations running on all nodes of the topology to which the OracleAS Guard client is connected, enter the following asgctl command at the asgctl prompt:

```
ASGCTL> show operation
**********************************
OPERATION: 19
  Status: running
  Elapsed Time: 0 days, 0 hours, 0 minutes, 28 secs
  TASK: syncFarm
    TASK: backupFarm
      TASK: fileCopyRemote
      TASK: fileCopyRemote
    TASK: restoreFarm
      TASK: fileCopyLocal
```

11.11.3 Displaying a List of Completed Operations

To display only operations that have completed (are *not* running on any nodes of the topology to which the OracleAS Guard client is connected for the current session), enter the following asgctl command at the asgctl prompt:

```
ASGCTL> show operation history
**********************************
OPERATION: 7
  Status: success
  Elapsed Time: 0 days, 0 hours, 0 minutes, 0 secs
  TASK: getTopology
    TASK: getInstance
**********************************
OPERATION: 16
  Status: success
  Elapsed Time: 0 days, 0 hours, 0 minutes, 0 secs
  TASK: getTopology
    TASK: getInstance
**********************************
OPERATION: 19
  Status: success
  Elapsed Time: 0 days, 0 hours, 1 minutes, 55 secs
  TASK: syncFarm
    TASK: backupFarm
      TASK: fileCopyRemote
      TASK: fileCopyRemote
    TASK: restoreFarm
      TASK: fileCopyLocall
```

11.11.4 Stopping an Operation

To stop a specific operation that is running on the server, enter the following asgctl command at the asgctl prompt and specify the operation number you want to stop. You can obtain the operation number you want to stop by entering a asgctl show operation full command.

```
ASGCTL> show operation full
**********************************
OPERATION: 19
  Status: running
```

```
  Elapsed Time: 0 days, 0 hours, 0 minutes, 28 secs
  Status: running
.
.
.
ASGCTL> stop operation 19
```

11.11.5 Tracing Tasks

To set a trace flag for a specific event and to log the output to the asgctl log files, enter the following asgctl command at the asgctl prompt and specify the **on** keyword and enter the trace flags to be enabled. In this case, the trace flag `DB` indicates that trace information regarding processing in the Oracle Database environment will be displayed. See the set trace command for more information about other trace flags that can be enabled. See the set trace command for a complete list of the trace flags that can be set.

```
ASGCTL> set trace on db
```

11.11.6 Writing Information About the Topology to a File

To write detailed information about the topology to which the local host is connected, enter the following asgctl command at the asgctl prompt and specify the path name and file name where the detailed output is to be written. The output is the same as the display shown in the dump topology command, except it is written to a file that you can save for future reference.

```
ASGCTL> dump topology to c:\dump_mid_1.txt
```

11.11.7 Error Messages

Appendix B, "OracleAS Guard Error Messages" categorizes and describes the error messages that may appear while using the OracleAS Disaster Recovery solution.

11.12 Wide Area DNS Operations

To direct client requests to the entry point of a production site, use DNS resolution. When a site switchover or failover is performed, client requests have to be redirected transparently to the new site that is playing the production role. To accomplish this redirection, the wide area DNS that resolves requests to the production site has to be switched over to the standby site. The DNS switchover can be accomplished by either using a wide area load balancer or manually changing DNS names.

> **Note:** A hardware load balancer is assumed to be front-ending each site. Check `http://metalink.oracle.com` for supported load balancers.

The following subsections describe the DNS switchover operation.

11.12.1 Using a Wide Area Load Balancer

When a wide area load balancer (global traffic manager) is deployed in front of the production and standby sites, it provides fault detection services and

performance-based routing redirection for the two sites. Additionally, the load balancer can provide authoritative DNS name server equivalent capabilities.

During normal operations, the wide area load balancer can be configured with the production site's load balancer name-to-IP mapping. When a DNS switchover is required, this mapping in the wide area load balancer is changed to map to the standby site's load balancer IP. This allows requests to be directed to the standby site, which now has the production role.

This method of DNS switchover works for both site switchover and failover. One advantage of using a wide area load balancer is that the time for a new name-to-IP mapping to take effect can be almost immediate. The downside is that an additional investment needs to be made for the wide area load balancer.

11.12.2 Manually Changing DNS Names

This method of DNS switchover involves the manual change of the name-to-IP mapping that is originally mapped to the IP address of the production site's load balancer. The mapping is changed to map to the IP address of the standby site's load balancer. Follow these instructions to perform the switchover:

1. Make a note the current time-to-live (TTL) value of the production site's load balancer mapping. This mapping is in the DNS cache and it will remain there until the TTL expires. As an example, let's assume that the TTL is 3600 seconds.
2. Modify the TTL value to a short interval (for example, 60 seconds).
3. Wait one interval of the original TTL. This is the original TTL of 3600 seconds from Step 1.
4. Ensure that the standby site is switched over to receive requests.
5. Modify the DNS mapping to resolve to the standby site's load balancer giving it the appropriate TTL value for normal operation (for example, 3600 seconds).

This method of DNS switchover works for planned site switchover operations only. The TTL value set in Step 2 should be a reasonable time period where client requests cannot be fulfilled. The modification of the TTL is effectively modifying the caching semantics of the address resolution from a long period of time to a short period. Due to the shortened caching period, an increase in DNS requests can be observed.

11.13 Using OracleAS Guard Command-Line Utility (asgctl)

This section includes the following subsections:

- Section 11.13.1, "Typical OracleAS Guard Session Using asgctl"
- Section 11.13.2, "Periodic Scheduling of OracleAS Guard asgctl Scripts"
- Section 11.13.3, "Submitting OracleAS Guard Jobs to the Enterprise Manager Job System"
- Section 11.14.1, "Special Considerations for Multiple OracleAS Metadata Repository Configurations"
- Chapter 12, "OracleAS Guard asgctl Command-line Reference"

11.13.1 Typical OracleAS Guard Session Using asgctl

A typical OracleAS Guard session using asgctl involves the following tasks, which are described in the following subsections:

- Section 11.13.1.1, "Getting Help"
- Section 11.13.1.2, "Specifying the Primary Database"
- Section 11.13.1.3, "Discovering the Topology"

One of the advantages of supporting an asgctl command-line interface is that you can place these asgctl commands in a proper sequence in a script as described in Section 11.13.1.4, "Creating and Executing an asgctl Script" and then execute the script as described in Section 11.13.2, "Periodic Scheduling of OracleAS Guard asgctl Scripts" and Section 11.13.3, "Submitting OracleAS Guard Jobs to the Enterprise Manager Job System".

11.13.1.1 Getting Help

To get help on a particular command, enter the asgctl command at the asgctl prompt and specify the command name you for which you want help information. Otherwise, to get help on all commands, enter the following asgctl command at the asgctl prompt:

```
ASGCTL> help
    connect asg [<host>] [ias_admin/<password>]
    disconnect
    exit
    quit
    clone topology to <standby_topology_host> [using policy <file>]
    clone instance <instance> to <standby_topology_host>
    discover topology [oidhost=<host>] [oidsslport=<sslport>] [oiduser=<user>]
oidpassword=<pass>
    discover topology within farm
    dump farm [to <file>]  (Deprecated)
    dump topology  [to <file>] [using policy <file>]
    dump policies
    failover [using policy <file>]
    help [<command>]
    instantiate farm to <standby_farm_host> (Deprecated)
    instantiate topology to <standby_topology_host> [using policy <file>]
    set asg credentials <host> ias_admin/<password> [for topology]
    set asg credentials <host> ias_admin/<password> [for farm] (Deprecated)
    set primary database <username>/<password>@<servicename> [pfile <filename> | spfile
<filename>]
    set new primary database <username>/<password>@<servicename> [pfile <filename> | spfile
<filename>]
    set noprompt
    set trace on|off <traceflags>
    sync farm to <standby_farm_host> [full | incr[emental]] (Deprecated)
    sync topology to <standby_topology_host> [full | incr[emental]] [using policy <file>]
    startup
    startup farm (Deprecated)
    startup topology
    shutdown [local]
    shutdown farm (Deprecated)
    shutdown topology
    show op[eration] [full] [[his]tory]
    show env
    stop op[eration] <op#>
    switchover farm to <standby_farm_host> (Deprecated)
    switchover topology to <standby_topology_host> [using policy <file>]
    verify farm [with <host>](Deprecated)
    verify topology [with <host>] [using policy <file>]
ASGCTL>
```

11.13.1.2 Specifying the Primary Database

To identify the OracleAS Infrastructure database on the primary topology, enter the following asgctl command at the asgctl prompt and specify the user name and password for the database account with sysdba privileges to access the OracleAS Infrastructure database and the TNS service name of the OracleAS Infrastructure database:

```
ASGCTL> set primary database sys/testpwd@asdb
Checking connection to database asdb
ASGCTL>
```

The standby site uses the same values as specified for the primary database because the service name and password for both the primary and standby OracleAS Infrastructure Databases must be the same. You must always set the primary database before performing an instantiate, sync, switchover, or failover operation.

If you have multiple OracleAS Metadata Repositories in your topology, you must authenticate each one using the set primary database command.

11.13.1.3 Discovering the Topology

You must perform a discover topology command when you first set up your OracleAS Disaster Recovery environment in order to initially create the topology XML file. There after, you should perform a discover topology operation whenever you procure another Oracle home in a production site or change roles from a production to a standby site through a switchover or failover operation. The discover topology command queries Oracle Internet Directory for all instances within the topology that share the same Oracle Internet Directory for the production site. Enter the following asgctl command at the asgctl prompt to discover the topology:

```
ASGCTL> discover topology oidpassword=oidpwd
Discovering topology on host "infra" with IP address "123.1.2.111" prodinfra:7890
    Connecting to the OID server on host "infra.us.oracle.com" using SSL port
"636" and username "orcladmin"
    Getting the list of databases from OID
    Gathering database information for SID "asdb" from host "infra.us.oracle.com"
    Getting the list of instances from OID
    Gathering instance information for "asr1012.infra.us.oracle.com" from host
"infra.us.oracle.com"
    Gathering instance information for "asmid1.asmid1.us.oracle.com" from host
"asmid1.us.oracle.com"
    Gathering instance information for "asmid2.asmid2.us.oracle.com" from host
"asmid2.us.oracle.com"
The topology has been discovered. A topology.xml file has been written to each
home in the topology.

ASGCTL>
```

After the production topology is known by OracleAS Guard for a production site, you can execute any one of the subsequent commands to perform a subsequent asgctl operation that involves the standby site. See discover topology for more information.

11.13.1.4 Creating and Executing an asgctl Script

To create a script containing a sequence of asgctl command names and their arguments, open an edit session with your favorite editor, enter the asgctl commands in the proper sequence according to the operations you want to perform, save the script file, then execute the script when you invoke asgctl as shown in the following command:

```
> ASGCTL @myasgctlscript.txt
```

See the set echo command for an example of a script containing a series of asgctl commands.

You can also set the noprompt state for use in executing commands in an asgctl script in which all interactive prompts are later ignored. See the asgctl set noprompt command for more information.

11.13.2 Periodic Scheduling of OracleAS Guard asgctl Scripts

For OracleAS Guard operations that you want to run periodically, such as a periodic sync topology operation to keep the standby topology synchronized with the primary topology, you can automate the periodic running of an OracleAS Guard asgctl script.

On UNIX systems, you can set up a cron job to run the asgctl script. Copy your asgctl script into the appropriate `/etc` subdirectory `cron.hourly`, `cron.daily`, `cron.weekly`, or `cron.monthly`. It will run either hourly, daily, weekly, or monthly, depending on the name of the subdirectory in which you choose to place your script. Or you can edit a crontab and create an entry that will be specific for the time on which you want to run the asgctl script. See the one or two manpages on cron and crontab for more information.

On Windows systems, you can use the task scheduler or scheduled tasks from the **Control Panel** to choose the time to run the asgctl script, daily, weekly, monthly, or at specific times. You can also purchase additional scheduler software with more options from a third party and then set the time and frequency to run the asgctl script. See the Windows operating system help for more information.

11.13.3 Submitting OracleAS Guard Jobs to the Enterprise Manager Job System

You can use the Enterprise Manager Job System to automate the execution of any asgctl script to be run at a specified time interval or at a specified time and date, or both, in addition to setting other custom settings. To do this, access the **EM Job Activity** page and create your own host command job to execute your asgctl script, which is called a job task. Your job task (script) will invoke asgctl to run the asgctl commands in the order in which they are listed. After you create your OracleAS Guard job, save it to the EM Job Library, which is a repository for frequently used jobs, where it can be executed based on the custom settings and time specifications you selected. See the Enterprise Manager online help and *Oracle Enterprise Manager Concepts* for more information.

11.14 Special Considerations for Some OracleAS Metadata Repository Configurations

This section describes special considerations for multiple OracleAS Metadata Repositories and OracleAS Metadata Repositories created using the OracleAS Metadata Repository Creation Assistant.

11.14.1 Special Considerations for Multiple OracleAS Metadata Repository Configurations

By default, the credentials you specified in the asgctl connect command are used whenever one OracleAS Guard server connects to another OracleAS Guard server. However, there may be cases where you want to do either of the following:

- Use different credentials for each system on a given site, see Section 11.14.1.1, "Setting asgctl Credentials".
- Use a common set of credentials in the standby topology that are the same as the credentials used in the primary topology, see Section 11.14.1.2, "Specifying the Primary Database".

If the credentials for any host system are not the same as those used in the asgctl connect command, you must set the OracleAS Guard credentials so that the OracleAS Guard server can connect to each host system in the configuration.

11.14.1.1 Setting asgctl Credentials

To set different credentials for all the host systems belonging to the same topology, enter the following asgctl command at the asgctl prompt. Specify the node name of the host system to which the credentials apply and the `ias_admin` account name and password for the `ias_admin` account created during the Oracle Application Server installation, and the key words **for topology**. These settings are good for the current session.

```
ASGCTL> set asg credentials standbyinfra ias_admin/<iasadminpwd> for topology
```

When you specify the key words, **for topology**, you set the credentials for all the host systems that belong to the same topology as the specified system; otherwise, the credentials will apply only for the specified host system.

The set asg credentials command is also useful when you want to use different credentials for a specific server on the topology. In the previous example, the same credentials were set for all nodes on the standby topology, so that these credentials differ from the credentials used in the primary topology. The following command sets the credentials for a specific node, the standbyinfra node, on the standby topology.

```
ASGCTL> set asg credentials standbyinfra ias_admin/<iasadminpwd>
```

To summarize, if you set the credentials for a topology, these credentials are inherited for the entire topology. If you set the credentials for an individual host on the topology, the credentials (for this host) override the default credentials set for the topology.

In addition, for topologies that have more than one Infrastructure, such as a collocated Oracle Internet Directory+OracleAS Metadata Repository and a separate Portal OracleAS Metadata Repository, OracleAS Guard requires that you set the credentials for each system on which an Infrastructure resides before performing any important OracleAS Guard operations, such as instantiate, sync, switchover, and failover. See set asg credentials for an example.

11.14.1.2 Specifying the Primary Database

To identify the OracleAS Infrastructure database on the primary topology, enter the following asgctl command at the asgctl prompt. Specify the user name and password for the database account with sysdba privileges to access the OracleAS Infrastructure Database on the primary topology and the TNS service name of the OracleAS Infrastructure database:

```
ASGCTL> set primary database sys/testpwd@asdb
Checking connection to database asdb
ASGCTL>
```

The standby site uses the same values as specified for the primary database because the service name and password for both the primary and standby OracleAS Infrastructure databases must be the same.

If a production or standby site has multiple OracleAS Metadata Repository instances installed and you are performing an instantiate, sync, switchover, or failover operation, you must identify all of the OracleAS Metadata Repository instances by performing a set primary database command for each OracleAS Metadata Repository instance before performing either an instantiate, sync, switchover, or failover operation. See set asg credentials for an example.

11.14.1.3 Setting OracleAS Guard Port Numbers

OracleAS Guard uses a default port (port) number of 7890; for example, `port=7890`. If there are any additional Oracle homes installed on a system, each additional Oracle home must have a unique OracleAS Guard port number, that is usually incremented by the value one, for example, `port=7891`, and so forth. See Section 11.4.6, "Supported OracleAS Disaster Recovery Configurations" for more information.

11.14.2 Special Considerations for OracleAS Metadata Repository Configurations Created Using OracleAS Metadata Repository Creation Assistant

The following items are special considerations for an OracleAS Metadata Repository configuration created using OracleAS Metadata Repository Creation Assistant. These Metadata Repository databases are installed in Oracle homes with schemas containing user data. For this reason, there are some special considerations regarding OracleAS Disaster Recovery.

- On the standby site, no Metadata Repository is created by OracleAS Disaster Recovery. The System Administrator must use the OracleAS Metadata Repository Creation Assistant on the standby site and create this Metadata Repository.
- During a clone topology operation to the standby site no instantiate operation is performed on the Metadata Repository.
- **Warning:** Do not perform a clone operation to a standby system containing an existing Oracle home because it will get overwritten. Only perform a clone operation to a standby system where there is no Oracle home installed.
- The OracleAS Disaster Recovery solution assumes that user schemas are already configured for Oracle Data Guard.
- The OracleAS Disaster Recovery solution assumes that when using Oracle Data Guard, that the Metadata Repository is not in managed recovery mode.
- OracleAS Disaster Recovery will not change the recovery mode of Oracle Data Guard for the Metadata Repository if it is found to be in managed recovery mode; instead, OracleAS Guard will issue a warning indicating that the database is in managed recovery mode and this feature must be set differently.
- OracleAS Guard must be installed in every Oracle home on every system that is part of your production and standby topology configured for the OracleAS Disaster Recovery solution. OracleAS Guard can be installed as a standalone install a kit located on OracleAS Companion CD #2. See the OracleAS Disaster Recovery installation information in *Oracle Application Server Installation Guide* for more information.

11.15 Special Considerations for OracleAS Disaster Recovery Environments

The following sections describe some additional special considerations for OracleAS Disaster Recovery environments.

11.15.1 Some Special Considerations That Must Be Taken When Setting Up Some OracleAS Disaster Recovery Sites

Some special considerations must be taken when setting up OracleAS Disaster Recovery for sites that include:

- Middle-tier CFC configurations
- OracleAS Guard release 10g (9.0.4) cloning

In both cases, the instance name stored in Oracle Internet Directory is comprised of the original host name on which the production site installation was performed. In the case of an OracleAS Disaster Recovery site having a symmetric topology, the standby OracleAS Disaster Recovery peer must be installed identically to the production site and for an OracleAS Guard Release 10.1.2.0.2 clone instance or clone topology operation, the operation must be performed to mirror the configuration.

In an asymmetric standby topology, where the production site physical host does not exist at the standby site, the instance name should be filtered out of the topology using the policy file capabilities (see Section 11.7, "Dumping Policy Files and Using Policy Files With Some asgctl Commands" for more information). The hosts file of the host on which a discover topology operation is performed must map the original host name to the corresponding IP of the new host system on which it was cloned.

11.15.2 Handling ons.conf and dsa.conf Configuration Files for Asymmetric Topologies

The OracleAS Guard operation synchronizes the configuration files of the standby site with those of the production site through a backup operation on the primary site and restores them to the standby site.

For asymmetric topologies the standby site has fewer nodes, thus node name list in the `ons.conf` configuration file is different from the one on the production site. Therefore, the `ons.conf` configuration file should be excluded from the backup list of files so it is not restored on the standby site. If not excluded, the nodes listed in the `ons.conf` configuration file will reflect the node list of the production site and not the actual node list of the standby site. This will cause inefficiencies as OPMN will continue to ping non existing nodes.

Additionally, for asymmetric topologies the `dsa.conf` configuration file for an Oracle home may contain special settings on the production site that are different from the standby site. For example, the `inventory_location` parameter setting may be different on the standby site than it is on the primary site. In this case, you should also exclude the `dsa.conf` configuration file from the backup list of files so it is not restored on the standby site. Otherwise, in this example, the location of the OraInventory will not be correct on the standby site following a switchover or failover operation.

In both these cases, you should modify the Backup and Restore exclusion file as follows to exclude both of these configuration files from the backup list of files so neither is then restored to the standby site:

```
# Exclude Files
# - Add additional files to this list that you want to be ignored
# - during the configuration file backup/restore
c:\oracle\ias1012\opmn\conf\ons.conf
c:\oracle\ias1012\dsa\dsa.conf
```

If the directives set in the `dsa.conf` file are necessary at the site that currently functions as the production site, it may be desirable to include the `dsa.conf` file for

synchronization and add a post switchover or failover step to edit physical site specific directives.

11.15.3 Other Special Considerations for OracleAS Disaster Recovery Environments

See Section 12.2, "Information Specific to a Small Set of OracleAS Guard Commands" for information describing some additional special considerations.

12

OracleAS Guard asgctl Command-line Reference

This chapter contains reference information describing the asgctl commands. Table 12–1 summarizes all the asgctl commands. Table 12–2 summarizes all the asgctl commands that were deprecated beginning with OracleAS release 10.1.2.0.2. Subsequent sections provide detailed reference information common to many commands and about each command.

Table 12–1 Summary of asgctl Commands

Command	Description
asgctl	Invokes the OracleAS Guard client command-line utility asgctl. On UNIX systems, `asgctl.sh` is located in *<ORACLE_HOME>*`/dsa/bin` and on Windows systems, `asgctl.bat` is located in *<ORACLE_HOME>*`\dsa\bin`.
clone instance	Clones a single production instance to a standby system.
clone topology	Clones two or more production middle tier instances to standby middle tier systems.
connect asg	Connects the OracleAS Guard client to the OracleAS Guard server.
disconnect	Disconnects the OracleAS Guard client from the OracleAS Guard server.
discover topology	Discovers by querying Oracle Internet Directory all instances within the topology that share the same Oracle Internet Directory for a production site and generates a topology XML file that describes the topology.
discover topology within farm	Discovers the topology within the farm for a site when Oracle Internet Directory is not available; in this case, OracleAS Guard server uses OPMN to discover the topology within the farm.
dump policies	Directs OracleAS Guard server to write detailed, default policy information to respective XML formatted files for a set of asgctl commands. Each policy file can then be edited and later specified to define the topology's disaster recovery policy to be used with the respective administrative command.
dump topology	Directs the OracleAS Guard server to write detailed information about the topology to the screen or if specified, to a file.
exit	Disconnects the OracleAS Guard client from any existing connections and exits the OracleAS Guard client. This has the same effect as the quit command.
failover	During an unscheduled outage of the production site, the standby site becomes the production site.

Table 12–1 (Cont.) Summary of asgctl Commands

Command	Description
help	Displays help information at the command line.
instantiate topology	Creates a topology at the standby site (after verifying that the primary and standby sites are valid for OracleAS Disaster Recovery); also synchronizes the standby site with the primary site so that the primary and standby sites are consistent.
quit	Disconnects the OracleAS Guard client from any existing connections and exits the OracleAS Guard client. This has the same effect as the exit command.
run	Remotely executes a script or program that resides in any home where OracleAS Guard is installed.
set asg credentials	Sets the credentials used to authenticate the OracleAS Guard client connections to OracleAS Guard servers and connections between OracleAS Guard servers to a specific host.
set echo	Sets command-echoing on or off in an asgctl script.
set new primary database	Identifies the OracleAS Infrastructure database on the standby topology as the new primary OracleAS Infrastructure database.
set noprompt	Sets the noprompt state in an asgctl script in which all interactive prompts are thereafter ignored.
set primary database	Identifies the OracleAS Infrastructure database on the primary topology.
set trace	Enables or disables tracing for the specified trace flag. When tracing for a flag is set to on, the output of the trace is written to the OracleAS Guard log files.
show env	Shows the current environment of the OracleAS Guard server to which the OracleAS Guard clients is connected.
show operation	Shows the current operation.
shutdown	Shuts down the OracleAS Guard server at the operating system command-line prompt on a system on which OPMN is not running. This command is only used with cloning an instance or cloning a topology.
shutdown topology	Shuts down a running topology.
startup	Starts up the OracleAS Guard server at the operating system command-line prompt on a system on which OPMN is not running. This command is only used with cloning an instance or cloning a topology.
startup topology	Starts up a shutdown topology.
stop operation	Stops the specified operation.
switchover topology	During a scheduled outage of the production site, switches the roles of the production site with the standby site so that the standby site now becomes the production site.
sync topology	Synchronizes the standby site with the primary site so that the primary and standby sites are consistent.
verify topology	Verifies that the topology is running and the configuration is valid. If a standby topology is specified, this command compares the primary and standby topologies to verify that they conform to the requirements for OracleAS Disaster Recovery.

Table 12–2 Summary of Deprecated asgctl Commands

Command	Description
dump farm (Deprecated)	Directs the OracleAS Guard server to write detailed information about the farm to the screen or if specified, to a file.
instantiate farm (Deprecated)	Creates a farm at the standby site (after verifying that the primary and standby sites are valid for OracleAS Disaster Recovery; also synchronizes the standby site with the primary site so that the primary and standby sites are consistent.
shutdown farm (Deprecated)	Shuts down a running farm.
startup farm (Deprecated)	Starts up a shutdown farm.
switchover farm (Deprecated)	During a scheduled outage of the production site, switches the roles of the production site with the standby site so that the standby site now becomes the production site.
sync farm (Deprecated)	Synchronizes the standby site with the primary site so that the primary and standby sites are consistent.
verify farm (Deprecated)	Verifies that the farm is running and the configuration is valid. If a standby farm is specified, this command compares the primary and standby farms to verify that they conform to the requirements for OracleAS Disaster Recovery.

12.1 Information Common to OracleAS Guard asgctl Commands

This section describes information that is common to OracleAS Guard asgctl commands.

General Information

The OracleAS Guard client must be connected to an OracleAS Guard server when you issue any asgctl command with the exception of startup and shutdown commands.

The OracleAS Guard server will act as the coordinating server for all operations performed on the systems being configured. By default, this is the local system where the `connect asg` command is being executed. This system must be a member of the production site topology.

OracleAS Guard Server Information

The OracleAS Guard server must be started on the standby host system (`<standby_topology_host>`. The OracleAS Guard server can be stopped and started using the opmnctl command-line Utility as follows:

```
On UNIX systems:
<ORACLE_HOME>/opmn/bin/opmnctl  startproc  ias-component=DSA

On Windows systems:
<ORACLE_HOME>\opmn\bin\opmnctl  stopproc  ias-component=DSA
```

12.2 Information Specific to a Small Set of OracleAS Guard Commands

This section describes information that is specific to a small set of OracleAS Guard operations, such as instantiate, sync, failover, switchover, dump topology, discover topology, clone topology, verify topology, setting the primary database, and setting asg credentials.

If a production or standby site has multiple OracleAS Metadata Repository instances installed and you are performing an instantiate, sync, switchover, or failover operation, you must identify all of the OracleAS Metadata Repository instances by performing a `set primary database` command for each and every OracleAS Metadata Repository instance prior to performing either an instantiate, sync, switchover, or failover operation.

OracleAS Guard requires that you set the credentials for any OracleAS Guard server system in the topology that has different credentials from the OracleAS Guard server to which you are connected before performing any important OracleAS Guard operations, such as instantiate, sync, switchover, and failover. See set asg credentials for an example.

You must perform a discover topology command when you first set up your OracleAS Disaster Recovery environment in order to initially create the topology XML file; there after, you should perform a discover topology operation whenever you procure another Oracle home in a production site or change roles from a production to a standby site through a switchover or failover operation.

If you want to use a policy file, edit the contents of the XML policy file to define by instance the domain of execution operations that are permitted for any one of these asgctl commands (clone topology, dump topology, failover, instantiate topology, switchover topology, sync topology, and verify topology). Each instance list entry in this XML policy file (`clone_policy.xml`, `dump_policy.xml`, `failover_policy.xml`, `instantiate_policy.xml`, `switchover_policy.xml`, `sync_policy.xml`, and `verify_policy.xml`) logically tags a production-standby peer combination with a particular attribute that defines the success requirement for the commands successful operation. See Section 11.7, "Dumping Policy Files and Using Policy Files With Some asgctl Commands" for more information and an example of an XML policy file.

12.2.1 Special Considerations for OracleAS Disaster Recovery Configurations in CFC Environments

In an OracleAS Disaster Recovery configuration that uses CFC on the primary topology or standby topology, or both, the following information must be considered before performing an asgctl clone, instantiate topology, switchover topology, or failover command. Before taking a cold backup or restoring the metadata repository database, the Oracle Backup and Recovery Tool shuts down the database first.

For example, in the Windows CFC environment, Oracle Fail Safe performs database polling and restarts the database if it is down. Hence, every time before the administrator performs a clone, instantiate, switchover, or failover operation, the administrator must disable database polling in Oracle Fail Safe and re-enable it after the backup/restore operation (after the clone, instantiate, switchover, or failover operation completes). The steps to perform this sequence of operations are described in a note in Section 12.2.1.1, "Special Considerations for Running Instantiate and Failover Operations in CFC Environments" and Section 12.2.1.3, "Special Considerations for Running a Switchover Operations in CFC Environments".

12.2.1.1 Special Considerations for Running Instantiate and Failover Operations in CFC Environments

In an OracleAS Disaster Recovery configuration that uses CFC on the primary topology or standby topology, or both, the following information must be considered before performing an asgctl clone, instantiate, switchover, or failover operation.

Before taking a cold backup or restoring the metadata repository database, the Oracle Backup and Recovery Tool shuts down the database first.

For example, in the Windows CFC environment, Oracle Fail Safe performs database polling and restarts the database if it is down. Hence, every time before the administrator performs an instantiate, switchover, or failover operation, the administrator must disable database polling in Oracle Fail Safe and re-enable it after the backup/restore operation (after the clone, instantiate, switchover, or failover operation completes).

The steps to perform this sequence of operations are as follows:

1. Using Microsoft Cluster Administrator, open the cluster group that contains the Application Server resources. Take the following resources offline in this order: Oracle Process Manager, then Oracle Database, then Oracle Listener.
2. Using Windows Service Control Manager, start the following services in this order: Fail Safe Listener, then the Oracle Database service.
3. From a Windows command prompt, use the sqlplus command-line Utility to startup the database.
4. Using Windows Service Control Manager, start the Oracle Process Manager.
5. Perform the asgctl commands, including the clone, instantiate, switchover, or failover operation.
6. Using Microsoft Cluster Administrator, open up the cluster group that contains the Application Server resources and bring up the following resources online in this order: Oracle Listener, then Oracle Database, then Oracle Process Manager.

12.2.1.2 A Special Consideration and Workaround for Performing an Instantiate Operation in CFC Environments

When performing an instantiate operation, OracleAS Guard puts an entry for the remote database in the `tnsnames.ora` file on both the production and standby site. The service name of this entry is constructed by concatenating _REMOTE1 to the database service name (for example, ORCL_REMOTE1). The entry contains the IP address of the target host where the database is running. On the production site, the IP will refer to the standby system and on the standby site, the IP refers to the production system.

In a CFC environment, the database is accessed using a virtual IP rather than a physical IP. When OracleAS Guard creates the `tnsnames.ora` entry it should use the virtual IP, but it uses the physical IP instead. This problem will be fixed in a future release of OracleAS Guard. As a workaround, when performing an instantiate operation in this environment, edit the `tnsnames.ora` file after an instantiation operation and replace the physical IP in the entry with the virtual IP used to access the database.

12.2.1.3 Special Considerations for Running a Switchover Operations in CFC Environments

In an OracleAS Disaster Recovery configuration that uses CFC on the primary topology or standby topology or both, the following information must be considered before performing an asgctl instantiate topology, switchover topology, or failover command.

Before taking a cold backup or restoring the metadata repository database, the Oracle Backup and Recovery Tool shuts down the database first.

For example, in the Windows CFC environment, Oracle Fail Safe performs database polling and restarts the database if it is down. Hence, every time before the administrator performs an instantiate, switchover, or failover operation, the administrator must disable database polling in Oracle Fail Safe and re-enable it after the backup/restore operation (after the instantiate, switchover, or failover operation completes).

The steps to perform this sequence of operations are as follows:

1. Using Microsoft Cluster Administrator, open the cluster group that contains the Application Server resources. Take the following resources offline in this order: Oracle Process Manager, then Oracle Database, then Oracle Listener.
2. Using Windows Service Control Manager, start the following services in this order: Fail Safe Listener, then the Oracle Database service.
3. From a Windows command prompt, use sqlplus to start up the database.
4. Perform the asgctl commands, including the instantiate topology, switchover topology, or failover command.
5. Using Microsoft Cluster Administrator, open up the cluster group that contains the Application Server resources and bring up the following resources online in this order: Oracle Listener, then Oracle Database, then Oracle Process Manager.

12.2.2 Other Special Considerations for OracleAS Disaster Recovery Environments

See Section 11.14, "Special Considerations for Some OracleAS Metadata Repository Configurations" and Section 11.15, "Special Considerations for OracleAS Disaster Recovery Environments" for information describing some additional special considerations for OracleAS Disaster Recovery environments.

asgctl

Invokes the OracleAS Guard client from the operating system command-line prompt or runs a script, if the path name to the script is provided.

Format

asgctl@[filename]

Parameters

filename = <file-path>
The path to a file that contains asgctl commands that you want to run as a script.

Usage Notes

On UNIX systems, `asgctl.sh` is located in *<ORACLE_HOME>*`/dsa/bin` and on Windows systems, `asgctl.bat` is located in *<ORACLE_HOME>*`\dsa\bin`.

Example

```
> asgctl.sh
Application Server Guard: Release 10.1.2.0.2

(c) Copyright 2004, 2005 Oracle Corporation. All rights reserved
ASGCTL>
```

clone instance

Clones a single production instance to a standby system.

Format

clone instance <instance> to <standby_topology_host>

Parameters

instance
The name of the instance.

standby_topology_host
The name of the standby topology host to which the instance is to be cloned.

Usage Notes

This command is useful for cloning a production instance on a middle tier to a standby middle tier host system. The clone instance operation eliminates the task of having to install the Oracle instance on the standby middle tier system and perform an instantiate operation.

The production instance to be cloned cannot exist on the standby system.

The following are prerequisites for performing the clone instance operation to the standby site system

- The OracleAS Guard standalone kit must be installed on the standby system.
- Backup and Restore must be installed in the OracleAS Guard home on the standby system
- A Java development kit with its jar utility must be installed on the standby system
- For Windows systems, the services kit (`sc.exe`) must be installed on the standby system

See Section 11.8, "OracleAS Guard Operations -- Standby Site Cloning of One or More Production Instances to a Standby System" for more information.

The basic procedure consists of the following pre-clone, clone, and post-clone steps.

Pre-Clone Steps

For each instance on the production and standby sites, perform the following steps:

1. Log in as su - root on UNIX systems or as Administrator on Windows systems.
2. CD to the instance home.
3. Shut down any OracleAS Guard servers.

```
On UNIX systems:
> <ORACLE HOME>/opmn/bin/opmnctl stopproc ias-component=DSA

On Windows systems:
C:\<ORACLE HOME>\opmn\bin\opmnctl stopproc ias-component=DSA
```

4. **On UNIX systems only:** make sure `dsaServer.sh` in *<ORACLE_HOME>*`/dsa/bin` is executable by everyone. If it is not, record the permission, then change the executable permission by issuing the following command:

```
chmod +x dsaServer.sh
chmod u+x asgexec
```

5. Invoke asgctl and issue the startup command.

```
>On UNIX systems from the <ORACLE_HOME>/dsa/bin directory
> asgctl.sh startup

On Windows systems fom the <ORACLE_HOME>\dsa\bin directory
C:\> asgctl startup
```

6. Log out as root on UNIX systems.

Clone Steps

From any instance on the production site, perform the following steps:

1. Log in as user (non root user on UNIX systems).
2. CD to the production instance home.
3. Invoke asgctl and run the clone instance command to clone the instance to the standby topology host system.
4. Log out of the system.

Post-Clone Steps

For the instance on the production and standby sites, perform the following steps:

1. Log in as su - root on UNIX systems or as Administrator on Windows systems.
2. CD to the instance home.
   - On the production site systems, CD to the instance home.
   - On the standby site systems, CD to the OracleAS Guard standalone home.
3. Perform an asgctl shutdown command.

```
>On UNIX systems from the <ORACLE_HOME>/dsa/bin directory
> asgctl.sh shutdown

On Windows systems fom the <ORACLE_HOME>\dsa\bin directory
C:\> asgctl shutdown
```

4. Log out as root on UNIX systems.
5. **On UNIX systems only:** Restore the permission for `dsaServer.sh` to what you recorded it as in Pre-Clone Step 4.
6. On the standby site only, CD to the newly cloned home.
7. Start up OracleAS Guard using the following opmnctl command:

```
On Unix systems:
> <ORACLE HOME>/opmn/bin/opmnctl startproc ias-component=DSA

On Windows systems:
C:\<ORACLE HOME>\opmn\bin\opmnctl startproc ias-component=DSA
```

> **Note:** If OracleAS Guard does not run as root on UNIX systems, the user will be prompted by the OracleAS Guard client to run the underlying operations at each of the instance homes as root (manually) in order to continue with the operation.

This last step completes the cloning instance operation and brings the systems back to where they were before you started the clone instance operation. At this point you could invoke asgctl, connect to a production system, discover the topology, and then perform a verify operation to determine whether the production and standby topologies were valid and consistent with one another as you would expect them to be.

Example

The following command in the example clones an instance named portal_2 to the standby topology host system named asmid2.

```
1. Check the prerequisites as described in the Usage Notes.
2. Perform the Pre-Clone steps as described in the Usage Notes.
3. Perform the Clone steps as described in the Usage Notes.
   a. Log in as user to any production system.
   b. CD to any production instance home.
   c. Invoke asgctl and perform the clone instance command.
> asgctl.sh
Application Server Guard: Release 10.1.2.0.2

(c) Copyright 2004, 2005 Oracle Corporation. All rights reserved
ASGCTL> connect asg prodinfra ias_admin/adminpwd
Successfully connected to prodinfra:7890
ASGCTL> set primary database sys/testpwd@asdb
Checking connection to database asdb
ASGCTL> clone instance portal_2 to asmid2
Generating default policy for this operation
.
.
.
ASGCTL> disconnect
ASGCTL> exit
>
   d. Log off the system
4. Perform Post-Clone steps as described in the Usage Notes.
```

clone topology

Clones two or more production middle tier instances to standby middle tier systems.

Format

clone topology to <standby_topology_host> [using policy <file>]

Parameters

standby_topology_host
The name of the standby topology host system.

using policy <file>
Full path and file specification for the XML policy file.

Usage Notes

This command is useful for cloning two or more production instances on middle tier systems to a standby middle tier host system. The clone topology operation eliminates the task of having to install these Oracle instances on the standby middle tier systems and perform an instantiate operation.

As part of the clone topology operation, the middle tiers are cloned and the OracleAS Metadata Repository is instantiated; however for a OracleAS Metadata Repository configuration created using OracleAS Metadata Repository Creation Assistant, no instantiate operation is performed.

The production instances to be cloned cannot exist on the standby systems.

The following are prerequisites for performing the clone topology operation to standby site systems.

- The OracleAS Guard standalone kit must be installed on each standby system
- Backup and Restore must be installed on each OracleAS Guard home on each standby system
- A Java development kit with its jar utility must be installed on each standby system
- For Windows systems only, the services kit (`sc.exe`) must be installed on each standby system

See Section 11.8, "OracleAS Guard Operations -- Standby Site Cloning of One or More Production Instances to a Standby System" for more information.

The basic procedure consists of the following pre-clone, clone, and post-clone steps.

Pre-Clone Steps

For each instance on the production and standby sites, perform the following steps:

1. Log in as su - root on UNIX systems or as Administrator on Windows systems.
2. CD to the instance home.
3. Shut down any OracleAS Guard servers.

```
On UNIX systems:
> <ORACLE HOME>/opmn/bin/opmnctl stopproc ias-component=DSA
```

```
On Windows systems:
C:\<ORACLE HOME>\opmn\bin\opmnctl stopproc ias-component=DSA
```

4. **On UNIX systems only:** make sure `dsaServer.sh` in *<ORACLE_HOME>*`/dsa/bin` is executable by everyone. If it is not, record the permission, then change the executable permission by issuing the following command:

```
chmod +x dsaServer.sh
chmod u+x asgexec
```

5. Invoke asgctl and issue the startup command.

```
>On UNIX systems from the <ORACLE_HOME>/dsa/bin directory
> asgctl.sh startup

On Windows systems fom the <ORACLE_HOME>\dsa\bin directory
C:\> asgctl startup
```

6. Log out as root on UNIX systems.

Clone Steps

From any instance on the production site, perform the following steps:

1. Log in as user (non root user on UNIX systems).
2. CD to any production instance home.
3. Invoke asgctl and run the clone topology command to clone the topology to the standby topology host system.
4. Log out of the system.

Post-Clone Steps

For each instance on the production and standby sites, perform the following steps:

1. Log in as su - root on UNIX systems or as Administrator on Windows systems.
2. CD to the instance home.
   - On the production site systems, CD to the instance home.
   - On the standby site systems, CD to the OracleAS Guard standalone home.
3. Perform an asgctl shutdown command.

```
>On UNIX systems from the <ORACLE_HOME>/dsa/bin directory
> asgctl.sh shutdown

On Windows systems fom the <ORACLE_HOME>\dsa\bin directory
C:\> asgctl shutdown
```

4. Log out as root on UNIX systems.
5. **On UNIX systems only:** Restore the permission for `dsaServer.sh` to what you recorded it as in Pre-Clone Step 4.
6. On the standby site only, CD to the newly cloned homes.
7. Start up OracleAS Guard using the following opmnctl command:

```
On Unix systems:
> <ORACLE HOME>/opmn/bin/opmnctl startproc ias-component=DSA

On Windows systems:
```

```
C:\<ORACLE HOME>\opmn\bin\opmnctl startproc ias-component=DSA
```

> **Note:** If OracleAS Guard does not run as root on UNIX systems, the user will be prompted by the OracleAS Guard client to run the underlying operations at each of the instance homes as root (manually) in order to continue with the operation.

This last step completes the cloning topology operation and brings the systems back to where they were before you started the clone topology operation. At this point you could invoke asgctl, connect to a production system, discover the topology, and then perform a verify operation to determine whether the production and standby topologies were valid and consistent with one another as you would expect them to be.

See Section 12.1, "Information Common to OracleAS Guard asgctl Commands" and Section 12.2, "Information Specific to a Small Set of OracleAS Guard Commands" for more information.

Example

The command in the following example results in the OracleAS Guard client cloning the topology to the standby topology host system standbyinfra.

```
1. Check the prerequisites as described in the Usage Notes.
2. Perform the Pre-Clone steps as described in the Usage Notes.
3. Perform the Clone steps as described in the Usage Notes.
   a. Log in as user to any production system.
   b. CD to any production instance Oracle home.
   c. Invoke asgctl and perform the clone instance command.
> asgctl.sh
Application Server Guard: Release 10.1.2.0.2

(c) Copyright 2004, 2005 Oracle Corporation. All rights reserved
ASGCTL> connect asg prodinfra ias_admin/adminpwd
Successfully connected to prodinfra:7890
ASGCTL> set primary database sys/testpwd@asdb
Checking connection to database asdb
ASGCTL> clone topology to standbyinfra
Generating default policy for this operation
.
.
.

# Command to use if you are using a policy file
# clone topology to standbyinfra using policy <file>
.
.
.
ASGCTL> disconnect
ASGCTL> exit
>
   d. Log off the system
4. Perform Post-Clone steps as described in the Usage Notes.
```

connect asg

Connects the OracleAS Guard client to the OracleAS Guard server on a system on which Oracle Application Server services are running.

Format

connect asg [<host-name>[:<port>]] ias_admin/<password>

Parameters

host-name = <host-name>
Name of the host system for the OracleAS Guard server to which you want the OracleAS Guard client to connect. This OracleAS Guard server will be the coordinating server for all operations performed on the systems being configured. The host name is optional if the OracleAS Guard client and OracleAS Guard server are on the same node.

port
The port number of the OracleAS Guard server in its Oracle home.

ias_admin/password
The user name must be the `ias_admin` account name and the password for the `ias_admin` account created during the Oracle Application Server installation.

Usage Notes

- The OracleAS Guard client system must have network access to the OracleAS Guard host system specified with the host-name parameter.
- The OracleAS Guard host system must have network access to all systems in the OracleAS Disaster Recovery configuration.
- The specified `ias_admin` account name must be configured with the necessary rights and privileges to permit OracleAS Disaster Recovery site operations (read and write access to all required files and directories, and so forth)
- An IP address can be used in place of a host name.
- If a password for the `ias_admin` account is not specified in the connect command, you will be prompted to enter a password.

Example

The command in the following example results in the OracleAS Guard client connecting to the OracleAS Guard server running on a host named prodinfra using the user name and password `ias_admin` and adminpwd, respectively.

```
ASGCTL> connect asg prodinfra ias_admin/adminpwd
Successfully connected to prodinfra:7890
```

disconnect

Disconnects the OracleAS Guard client from the OracleAS Guard server to which it is currently connected.

Format

disconnect

Usage Notes

The OracleAS Guard client must be connected to a OracleAS Guard server when you issue this command.

Example

The command in the following example disconnects the OracleAS Guard client from the OracleAS Guard server to which it is currently connected.

```
ASGCTL> disconnect
ASGCTL>
```

discover topology

Directs asgctl to query Oracle Internet Directory and determine all instances within the topology that share the same Oracle Internet Directory for a production site and generates a topology XML file that describes the topology.

Format

discover topology [oidhost=<host>] [oidsslport=<sslport>] [oiduser=<user>] oidpassword=<pass>

Parameters

host
Name of the host system where Oracle Internet Directory is installed.

sslport
The port number of the host system where Oracle Internet Directory and Secure Sockets Layer (SSL) is installed.

user
The Oracle Internet Directory user name.

pass
The password for the specified Oracle Internet Directory user name.

Usage Notes

You should perform a discover topology operation whenever you procure another Oracle home in a production site or change roles from a production to a standby site through a switchover or failover operation.

Discover topology creates the topology (stored in `topology.xml`) on which to perform all OracleAS Guard operations. This command utilizes the information in Oracle Internet Directory to define the instances included in the topology. Additionally, it gathers local information about each instance. For this reason, it requires all production site instances to have OPMN running. For instances not managed using a DCM farm, the OracleAS Guard service on the Oracle home has to be started. If the services are not started locally, a warning will be produced and the `topology.xml` file will contain only the instances discovered.

See Section 12.1, "Information Common to OracleAS Guard asgctl Commands" and Section 12.2, "Information Specific to a Small Set of OracleAS Guard Commands" for more information.

Example

The command in the following example discovers all the instances within the topology that share the same Oracle Internet Directory for a production site, and generates a topology XML file that describes the topology.

```
ASGCTL> connect asg prodinfra ias_admin/adminpwd
Successfully connected to prodinfra:7890
ASGCTL> discover topology oidpassword=oidpwd
Discovering topology on host "infra" with IP address "123.1.2.111" prodinfra:7890
    Connecting to the OID server on host "infra.us.oracle.com" using SSL port
"636" and username "orcladmin"
    Getting the list of databases from OID
```

```
    Gathering database information for SID "asdb" from host "infra.us.oracle.com"
    Getting the list of instances from OID
    Gathering instance information for "asr1012.infra.us.oracle.com" from host
"infra.us.oracle.com"
    Gathering instance information for "asmid1.asmid1.us.oracle.com" from host
"asmid1.us.oracle.com"
    Gathering instance information for "asmid2.asmid2.us.oracle.com" from host
"asmid2.us.oracle.com"
The topology has been discovered. A topology.xml file has been written to each
home in the topology.
ASGCTL>
```

discover topology within farm

Directs asgctl to discover the topology within a farm at a production site for those special cases where a farm does not have Oracle Internet Directory available.

> **Note:** You should always use the discover topology command for discovering the topology for a site because this command uses Oracle Internet Directory to discover all instances in the topology. The discover topology within farm command is useful only in those special cases where Oracle Internet Directory is not available; in this special case OracleAS Guard uses OPMN to discover the topology within a farm.

Format

discover topology within farm

Parameters

None.

Usage Notes

The OracleAS Guard client must be connected to a OracleAS Guard server when you issue this command.

Example

The command in the following example for a special case in which Oracle Internet Directory is not available, uses OPMN to discover the application server topology within a farm of the OracleAS Guard server to which the OracleAS Guard client is currently connected.

```
ASGCTL> connect asg prodinfra ias_admin/adminpwd
Successfully connected to prodinfra:7890
ASGCTL> set primary database sys/testpwd@asdb
Checking connection to database asdb
ASGCTL> discover topology within farm
Warning: If OID is part of your environment, you should use it for discovery
Discovering topology on host "infra" with IP address "123.1.2.111"
prodinfra:7890
    Discovering instances within the topology using OPMN
    Gathering instance information for "asr1012.infra.us.oracle.com" from host
"infra.us.oracle.com"
The topology has been discovered. A topology.xml file has been written to each
home in the topology.
ASGCTL>
```

dump policies

Directs OracleAS Guard Server to write detailed, default policy information in XML formatted output for the different asgctl commands to a set of policy files located on the local host at the *<ORACLE_HOME>*`/dsa/conf` directory on UNIX systems or *<ORACLE_HOME>*`\dsa\conf` directory on Windows systems.

Format

dump policies

Parameters

None.

Usage Notes

A set of XML formatted policy files are written for each of the following asgctl commands: clone topology, dump topology, failover, instantiate topology, sync topology, switchover topology, and verify topology. You can edit the respective command's policy file, then specify it in the `using policy <file>` clause for the appropriate command. This parameter lets you define the topology's disaster recovery policy for each of these OracleAS Guard operations.

For the dump policy file, by default the success requirement attribute is set to optional for all instances (middle tier and OracleAS Metadata Repository).

For the failover policy file, by default the success requirement attribute is set to optional for all instances (middle tier and OracleAS Metadata Repository) and mandatory for the Oracle Internet Directory home.

For the instantiate policy file, by default the success requirement attribute is set to mandatory for all instances.

For the switchover policy file, by default the success requirement attribute is set to optional for all instances (middle tier and OracleAS Metadata Repository) and mandatory for the Oracle Internet Directory home.

For the sync policy file, by default the success requirement attribute is set to mandatory for all instances.

For the verify policy file, by default the success requirement attribute is set to optional for all instances (middle tier and OracleAS Metadata Repository) and mandatory for the Oracle Internet Directory home.

Example

The following example writes detailed, default policy information in XML formatted output for the different asgctl commands to a set of respective policy files located on the local host.

```
ASGCTL> dump policies
Generating default policy for this operation
Creating policy files on local host in directory
"/private1/OraHome2/asr1012/dsa/conf/"
ASGCTL>
```

dump topology

Directs asgctl to write detailed information about the topology to the specified file.

Format

dump topology [to <file>] [using policy <file>]

Parameters

to <file>
Name of file on the OracleAS Guard client node where the detailed output is to be written.

using policy <file>
Full path and file specification for the XML policy file.

Usage Notes

For the dump policy file, by default the success requirement attribute is set to optional for all OracleAS homes (middle tier and OracleAS Metadata Repository).

Example

The following example writes detailed information about the topology to a local file.

```
ASGCTL> connect asg prodinfra ias_admin/adminpwd
Successfully connected to prodinfra:7890
ASGCTL> set primary database sys/testpwd@asdb
Checking connection to database asdb
ASGCTL> dump topology to c:\dump_mid_1.txt

Contents of file c:\dump_mid_1.txt are:

Generating default policy for this operation

 Instance: asr1012.infra.us.oracle.com
    Type: Infrastructure
    Oracle Home Name: asr1012
    Oracle Home Path: /private1/OraHome
    Version: 10.1.2.0.2
    OidHost: infra.us.oracle.com
    OidPort: 389
    VirtualHost: infra.us.oracle.com
    Host: prodinfra
    Ip: 123.1.2.111
    Operation System Arch: sparc
    Operation System Version: 5.8
    Operation System Name: SunOS

 Instance: asmid2.asmid2.us.oracle.com
    Type: Core
    Oracle Home Name: asmid2
    Oracle Home Path: /private1/OraHome2
    Version: 10.1.2.0.2
    OidHost: infra.us.oracle.com
    OidPort: 389
    VirtualHost: asmid2.us.oracle.com
```

```
    Host: asmid2
    Ip: 123.1.2.333
    Operation System Arch: sparc
    Operation System Version: 5.8
    Operation System Name: SunOS

 Instance: asmid1.asmid1.us.oracle.com
    Type: Core
    Oracle Home Name: asmid1
    Oracle Home Path: /private1/OraHome
    Version: 10.1.2.0.2
    OidHost: infra.us.oracle.com
    OidPort: 389
    VirtualHost: asmid1.us.oracle.com
    Host: asmid1
    Ip: 123.1.2.334
    Operation System Arch: sparc
    Operation System Version: 5.8
    Operation System Name: SunOS
ASGCTL>
```

The following example writes detailed information about the topology to a local file. Any instances that you want left out of the output can be specified in the policy file.

```
# Command to use if you are using a policy file
ASGCTL> dump topology to c:\dump_mid_1.txt using policy <file>
```

exit

Disconnects from any existing connections to OracleAS Guard servers and exits from the OracleAS Guard client.

Format

exit

Parameters

None

Usage Notes

None.

Example

```
ASGCTL> exit
>
```

failover

During an unscheduled outage of the production site, performs the failover operation on the standby site to make it the primary site.

Format

failover [using policy <file>]

Parameters

using policy file
Full path and file specification for the XML policy file.

Usage Notes

Make sure OracleAS Infrastructure database is running on the standby topology before performing a failover operation. Also, the OracleAS Infrastructure database information must be set by using the set new primary database asgctl command.

The global DNS names are used to direct the failover. This will be different than the HA naming utilized in the OracleAS Disaster Recovery environment. The discovery mechanism automatically maps the topology to the corresponding peer, based off local name resolution.

For the failover policy file, by default the success requirement attribute is set to optional for all OracleAS homes (middle tier and OracleAS Metadata Repository) and mandatory for the Oracle Internet Directory home.

See Section 12.1, "Information Common to OracleAS Guard asgctl Commands" and Section 12.2, "Information Specific to a Small Set of OracleAS Guard Commands" for more information.

Example

The following example performs a failover operation to a standby site.

```
ASGCTL> connect asg standbyinfra ias_admin/adminpwd
Successfully connected to standbyinfra:7890
ASGCTL> set new primary database sys/testpwd@asdb
ASGCTL> failover
Generating default policy for this operation
standbyinfra:7890
    Failover each instance in the topology from standby to primary topology
standbyinfra:7890 (home /private1/OraHome2/asr1012)
    Shutting down each instance in the topology
.
.
.
    Executing opmnctl startall command
standbyinfra:7890
     HA directory exists for instance asr1012.infra.us.oracle.com
asmid2:7890
     HA directory exists for instance asmid2.asmid2.us.oracle.com
asmid1:7890
     HA directory exists for instance asmid1.asmid1.us.oracle.com
ASGCTL>
```

```
# Command to use if you are using a policy file
# failover using policy <file>
```

help

Displays help information.

Format

help [<command>]

Parameters

command
Name of the command for which you want help.

Usage Notes

None.

Example

The following example displays help about all commands.

```
ASGCTL> help
    connect asg [<host>] [ias_admin/<password>]
    disconnect
    exit
    quit
    clone topology to <standby_topology_host> [using policy <file>]
    clone instance <instance> to <standby_topology_host>
    discover topology [oidhost=<host>] [oidsslport=<sslport>] [oiduser=<user>] oidpassword=<pass>
    discover topology within farm
    dump farm [to <file>]  (Deprecated)
    dump topology  [to <file>] [using policy <file>]
    dump policies
    failover [using policy <file>]
    help [<command>]
    instantiate farm to <standby_farm_host> (Deprecated)
    instantiate topology to <standby_topology_host> [using policy <file>]
    set asg credentials <host> ias_admin/<password> [for topology]
    set asg credentials <host> ias_admin/<password> [for farm] (Deprecated)
    set primary database <username>/<password>@<servicename> [pfile <filename> | spfile <filename>]
    set new primary database <username>/<password>@<servicename> [pfile <filename> | spfile <filename>]
    set noprompt
    set trace on|off <traceflags>
    sync farm to <standby_farm_host> [full | incr[emental]] (Deprecated)
    sync topology to <standby_topology_host> [full | incr[emental]] [using policy <file>]
    startup
    startup farm (Deprecated)
    startup topology
    shutdown [local]
    shutdown farm (Deprecated)
    shutdown topology
    show op[eration] [full] [[his]tory]
    show env
    stop op[eration] <op#>
    switchover farm to <standby_farm_host> (Deprecated)
    switchover topology to <standby_topology_host> [using policy <file>]
    verify farm [with <host>](Deprecated)
    verify topology [with <host>] [using policy <file>]
ASGCTL>
```

instantiate topology

Instantiates a topology to a standby site by establishing the relationship between standby and production instances, mirroring the configuration, creating the standby Infrastructure, and then synchronizing the standby site with the primary site.

Format

instantiate topology to <standby_topology_host>[:<port>] [with cloning] [using policy <file>]

Parameters

standby_topology_host
Name of the standby host system. This parameter is required because it directs the coordinating OracleAS Guard server instance to discover the instances that make up the standby site. This host system must be a member of the standby topology.

port
The port number of the OracleAS Guard server in its Oracle home.

with cloning
A directive to perform an instantiation operation using cloning.

using policy <file>
Full path and file specification for the XML policy file.

Usage Notes

Make sure OracleAS Infrastructure database is running on the primary topology before performing an instantiate topology operation. Also, the OracleAS Infrastructure database information must be set by using the set primary database asgctl command.

The global DNS names are used to direct the instantiation. This will be different than the HA naming utilized in the OracleAS Disaster Recovery environment. The discovery mechanism automatically maps the topology to the corresponding peer, based off local name resolution.

The instantiate operation performs an implicit verify operation.

For the instantiate policy file, by default the success requirement attribute is set to mandatory for all instances.

See Section 12.1, "Information Common to OracleAS Guard asgctl Commands" and Section 12.2, "Information Specific to a Small Set of OracleAS Guard Commands" for more information.

Example

The following example instantiates a standby topology by attaching the coordinating OracleAS Guard server and discovering the topology of the production and standby sites, performing site verification, and establishing a OracleAS Disaster Recovery environment with the topology containing the standby topology host known by DNS as standbyinfra. Note that part way through the operation you will be prompted to answer a question regarding whether you want to shut down the database. Reply by entering `y` or `yes`.

```
ASGCTL> connect asg prodinfra ias_admin/adminpwd
Successfully connected to prodinfra:7890
```

```
ASGCTL> set primary database sys/testpwd@asdb
Checking connection to database asdb
ASGCTL> instantiate topology to standbyinfra
Generating default policy for this operation
prodinfra:7890
    Instantiating each instance in the topology to standby topology
     HA directory exists for instance asr1012.infra.us.oracle.com
asmid2:7890
     HA directory exists for instance asmid2.asmid2.us.oracle.com
asmid1:7890
     HA directory exists for instance asmid1.asmid1.us.oracle.com
standbyinfra:7890
     HA directory exists for instance asr1012.infra.us.oracle.com
asmid2:7890
     HA directory exists for instance asmid2.asmid2.us.oracle.com
asmid1:7890
     HA directory exists for instance asmid1.asmid1.us.oracle.com
asmid2:7890
    Verifying that the topology is symmetrical in both primary and standby configuration
.
.
.
This operation requires the database to be shutdown. Do you want to continue? Yes or No
y
.
.
.
asmid2:7890 (home /private1/oracle/asr1012)
    Starting backup/synchronization of database "orcl.us.oracle.com"
    Starting restore/synchronization of database "orcl.us.oracle.com"
    Synchronizing topology completed successfully
asmid2:7890
    Synchronizing topology completed successfully

ASGCTL>

# Command to use if you are using a policy file
# instantiate topology to standbyinfra using policy <file>
```

quit

Instructs the OracleAS Guard client to disconnect from any existing connections and exit from asgctl.

Format

quit

Parameters

None

Usage Notes

None.

Example

The following example exits from asgctl.

```
ASGCTL> quit
>
```

run

Remotely executes a script or program that resides in any home where OracleAS Guard is installed. The run command can be executed within a topology or at the specified instance.

Format

run [at topology [using policy <file>]] <command>

run [at instance <instance_name>] <command>

Parameters

at topology
A keyword, that if present in the command line, directs OracleAS Guard to perform the run operation across the topology.

using policy <file>
Full path and file specification for the XML policy file.

command
The name as a command string of the script or binary program to be executed.

at instance
A keyword, that if present in the command line, directs OracleAS Guard to perform the run operation at the specified instance.

instance_name
The name of the instance where the run command is to be executed.

Usage Notes

This command is useful for remotely executing a script or program that resides in any Oracle home where OracleAS Guard is installed. The script or program must be physically located in each Oracle home across the topology or at that specified instance where it is expected to be executed. The asgctl user must first connect to the OracleAS Guard server specifying the Application Server JAZN credentials (ias_admin or oc4jadmin) before invoking this asgctl run command. It is assumed that if the user knows the JAZN credentials, then the user should be allowed to execute a script or program in the home. Upon receiving a run command invocation, OracleAS Guard will verify that the file specified in the command string exists in the Oracle home where the OracleAS Guard server is running before executing the script or program. The output of the script is echoed back to the asgctl console.

Example

For Oracle RAC DisasterRecovery deployments, shut down all instances prior to the switchover operation. To do this, create a script and then execute the script using the run command. See Section 11.10.1.1, "Scheduled Outages", step 5 for details. The following command in this example assumes a script is written and then remotely runs the script `shutdown_asdb_instance.sh` for the instance named asdb. This script utilizes the ASG distributed ASG scripting capabilities and allows a system administrator to perform the switchover operation from within the asgctl utility.

```
ASGCTL> run at instance asdb shutdown_asdb_instance.sh
```

set asg credentials

Sets the credentials used to authenticate the OracleAS Guard connections to OracleAS Guard servers.

Format

set asg credentials <host>[:<port>] ias_admin/<password> [for farm] [for topology]

Parameters

host
Name of the host system to which the credentials apply. When OracleAS Guard connects to that host, it will use these credentials.

port
The port number of the OracleAS Guard server in its Oracle home.

ias_admin/password
The user name must be the `ias_admin` account name and the password for the `ias_admin` account created during the Oracle Application Server installation. This account name must be the same as the account name on at least one of the Oracle Application Server homes.

for farm (deprecated)
A keyword, that if present in the command line, directs OracleAS Guard to set the credentials for all of the host systems that belong to the same farm as the local host system.

for topology
A keyword, that if present in the command line, directs OracleAS Guard to set the credentials for all of the host systems that belong to the same topology as the local host system.

Usage Notes

By default, the credentials used in the asgctl connect command are used whenever a OracleAS Guard server needs to connect to another OracleAS Guard server. However, there may be cases where you want to use different credentials for a specific server. This command allows you to use the same credentials for all nodes in a topology. For example, you may want to use a common set of credentials in the standby topology that is different from the credentials used in the primary topology.

If you set the credentials for a topology, these credentials are inherited for the entire topology. If you set the credentials for an individual host on the topology, the credentials (for this host) override the default credentials set for the topology.

For topologies that have more than one Infrastructure, such as a collocated Oracle Internet Directory+OracleAS Metadata Repository and a separate Portal OracleAS Metadata Repository, OracleAS Guard requires that you set the credentials for each system on which an Infrastructure resides before performing any important OracleAS Guard operations, such as instantiate, sync, switchover, and failover. This is actually a two step process in which you must first identify all OracleAS Infrastructure databases on the topology using the set the primary database command for each Infrastructure, then you must set the credentials used to authenticate the OracleAS Guard connections to OracleAS Guard servers on which these Infrastructures reside. The

following example illustrates this concept. Assume your production topology and standby topology consists of the following systems with installed Infrastructure and middle tier software applications.

Production topology:

host01 (Identity Management+OracleAS Metadata Repository), host04 (OracleAS Metadata Repository only), host06 (J2EE), host06 (Portal & Wireless)

Standby Topology:

host02 (Identity Management+OracleAS Metadata Repository), host05 (OracleAS Metadata Repository only), host07 (J2EE), host07 (Portal & Wireless)

The following OracleAS Guard set primary database and set asg credentials commands would be required to properly identify the Infrastructures and authenticate OracleAS Guard connections to OracleAS Guard servers prior to performing an instantiate, sync, switchover, or failover operation. Assuming that the Oracle Identity Management+OracleAS Metadata Repository Infrastructure has a service name of `orcl` and the separate Portal OracleAS Metadata Repository has a service name of `asdb`.

```
ASGCTL> set primary database sys/<password>@orcl.us.oracle.com
ASGCTL> set primary database sys/<password>@asdb.us.oracle.com
ASGCTL> set asg credentials host01.us.oracle.com ias_admin/<password>
ASGCTL> set asg credentials host04.us.oracle.com ias_admin/<password>
```

Note that for a failover operation, these steps would be carried out on the standby topology and are as follows with a change in the host system names:

```
ASGCTL> set primary database sys/<password>@orcl.us.oracle.com
ASGCTL> set primary database sys/<password>@asdb.us.oracle.com
ASGCTL> set asg credentials host02.us.oracle.com ias_admin/<password>
ASGCTL> set asg credentials host05.us.oracle.com ias_admin/<password>
```

The OracleAS Guard client must be connected to a OracleAS Guard server before using this command.

An IP address can be used in place of a host name.

See Section 12.1, "Information Common to OracleAS Guard asgctl Commands" and Section 12.2, "Information Specific to a Small Set of OracleAS Guard Commands" for more information.

Example

The following example sets the OracleAS Guard credentials of host system standbyinfra to all host systems that belong to this topology.

```
ASGCTL> set asg credentials standbyinfra ias_admin/<password> for topology
```

set echo

Sets command-echoing on or off in a asgctl script.

Format

set echo on | off

Parameters

on | off
Specifying "on" turns on command-echoing in a asgctl script. Specifying "off" turns off command-echoing in a asgctl script.

Usage Notes

This command is useful when running large asgctl scripts. For example, if the asgctl script has error test cases with comments entered before each test case or before each asgctl command, setting echo on displays the comment before each test case or before each asgctl command that is run to give you an explanation of what the test case is or what asgctl command is about to be run.

This command also works with nested scripts.

Example

The following example is a asgctl script that turns on command-echoing, runs a test case, connects to a OracleAS Guard server, displays detailed information about the topology, then turns echo off, disconnects from the OracleAS Guard server, and exits from the OracleAS Guard client.

```
> ASGCTL @myasgctltestscript.txt

# myasgctltestscript.txt
# turn on echo
set echo on

# make sure you are not connected
disconnect

# not connected, should get an error message
dump topology

# connect to a DSA server
connect asg prodinfra ias_admin/adminpwd

#display detailed info about the topology
dump topology

#disconnect
disconnect

# turn off echo
echo off
exit
```

set new primary database

Identifies the OracleAS Infrastructure database on the standby topology as the new primary database preceding a failover operation. This command is only used as part of a failover operation.

Format

set new primary database <username>/<password>@<servicename> [pfile <filename> | spfile <filename>]

Parameters

username/password
User name and password for the database account with sysdba privileges.

servicename
The TNS service name of the OracleAS Infrastructure database. The name must be defined on the OracleAS Infrastructure host system; it does not need to be defined on the OracleAS Guard client host system.

pfile filename
The filename of the primary (OracleAS Infrastructure) database initialization file that will be used when the primary database is started.

spfile filename
The filename of the server (OracleAS Infrastructure) initialization file that will be used when the database is started.

Usage Notes

Before performing a failover operation, you are required to connect to the Infrastructure node of the standby topology and define the new primary database. Once the Oracle Infrastructure database on the standby site is identified as the new primary database, then you can proceed to begin the failover operation.

Example

The following example sets the OracleAS Infrastructure database information for the standby topology as the new primary/production topology preceding a failover operation.

```
ASGCTL> connect asg standbyinfra ias_admin/adminpwd
Successfully connected to standbyinfra:7890
ASGCTL> set new primary database sys/testpwd@asdb
ASGCTL> failover
.
.
.
ASGCTL>
```

set noprompt

Sets the noprompt state for user interaction for use in executing commands in an asgctl script.

Format

set noprompt

Parameters

None

Usage Notes

The default value, if supplied, is taken for all interactive prompts. A prompt for a user name and password returns an error message in the noprompt state.

Example

The following example is an asgctl script containing an asgctl set noprompt command part way through the script that thereafter ignores all subsequent interactive prompting.

```
> ASGCTL @myasgctltestscript.txt

# myasgctltestscript.txt

# connect to a DSA server
connect asg prodinfra ias_admin/adminpwd

# set the primary database
set primary database sys/testpwd@asdb

# discover the production topology
discover topology oidpassword=oidpwd

# set the noprompt state
set noprompt

#display detailed info about the topology
dump topology

#disconnect
disconnect

exit
```

set primary database

Identifies the OracleAS Infrastructure database on the primary topology.

Format

set primary database <username>/<password>@<servicename> [pfile <filename> | spfile <filename>]

Parameters

username/password
User name and password for the database account with sysdba privileges.

servicename
The TNS service name of the OracleAS Infrastructure database. The name must be defined on the OracleAS Infrastructure host system; it does not need to be defined on the OracleAS Guard client host system.

pfile filename
The filename of the primary (OracleAS Infrastructure) database initialization file that will be used when the primary database is started.

spfile filename
The filename of the server (OracleAS Infrastructure) initialization file that will be used when the database is started.

Usage Notes

You must always set the primary database before performing an instantiate, sync, or switchover operation.

When you set the primary database, OracleAS Guard server logs into and validates the connection to the database.

If a production or standby site has multiple OracleAS Metadata Repository instances installed and you are performing an instantiate, sync, switchover, or failover operation, you must identify all of the OracleAS Metadata Repository instances by performing a set primary database command for each and every OracleAS Metadata Repository instance prior to performing either an instantiate, sync, switchover, or failover operation. In addition, for topologies that have more than one Infrastructure, such as a collocated Oracle Internet Directory+OracleAS Metadata Repository and a separate Portal OracleAS Metadata Repository, OracleAS Guard requires that you set the credentials for each system on which an Infrastructure resides before performing any important OracleAS Guard operations, such as instantiate, sync, switchover, and failover. See set asg credentials for an example.

OracleAS Guard requires the database to have password file authentication. If the database does not have a password file, you must use the `orapwd` utility to create a password file. Also, set the `REMOTE_LOGIN_PASSWORDFILE` initialization parameter to `EXCLUSIVE`.

See Section 12.1, "Information Common to OracleAS Guard asgctl Commands" and Section 12.2, "Information Specific to a Small Set of OracleAS Guard Commands" for more information.

Example

The following example sets the OracleAS Infrastructure database information for the primary or production topology.

```
ASGCTL> connect asg prodinfra ias_admin/adminpwd
Successfully connected to prodinfra:7890
ASGCTL> set primary database sys/testpwd@asdb
Checking connection to database asdb
ASGCTL>
```

The following example sets OracleAS Infrastructure database information for each OracleAS Metadata Repository installed for the primary/production topology prior to a switchover operation.

```
ASGCTL> connect asg prodinfra ias_admin/adminpwd
Successfully connected to prodinfra:7890
ASGCTL> set primary database sys/testpwd@portal_1
Checking connection to database portal_1
ASGCTL> set primary database sys/testpwd@portal_2
Checking connection to database portal_2
ASGCTL> set primary database sys/testpwd@asdb
Checking connection to database asdb
ASGCTL> discover topology oidpassword=oidpwd
ASGCTL> switchover topology to standbyinfra
.
.
.
```

set trace

Sets a trace flag on or off to log output to the OracleAS Guard log files.

Format

set trace on | off <traceflags>

Parameters

on | off
Specifying "on" enables tracing. Specifying "off" disables tracing.

traceflags
The traceflags to be enabled. Two or more specified traceflags entries must be separated by a comma (,). The traceflags are as follows:

- DB -- trace information regarding processing in the Oracle Database environment
- HOME -- trace information with regard to Oracle homes
- IAS -- trace information regarding processing in Oracle Application Server
- OPMN -- trace information regarding access to OracleAS OPMN calls
- IP -- trace information regarding network access and address translation
- CLIPBOARD -- trace information regarding clipboard processing
- COPY -- trace information regarding file copy processing
- FLOW -- trace information regarding work flow processing
- NET -- trace information regarding network processing
- RUNCMD -- trace information regarding the running of external commands
- SESSION -- trace information regarding session management
- TOPOLOGY -- trace information regarding processing of topology information

Usage Notes

This command applies to all hosts that might be involved in a asgctl command during the lifetime of the connection.

The OracleAS Guard client must be connected to a OracleAS Guard server before using this command.

Example

The following example turns on trace for database operations.

```
ASGCTL> set trace on db
```

show env

Shows the current environment for the OracleAS Guard server to which the OracleAS Guard client is connected.

Format

show env

Parameters

None.

Usage Notes

None.

Example

The following examples show the environment of the OracleAS Guard server to which the OracleAS Guard client is connected. In the first example, the primary database and new primary database are not yet set on host prodinfra and in the second example, the primary database has already been set on host standbyinfra.

Example 1.

```
ASGCTL> show env

     ASG Server Connection:
        Host: prodinfra
        Port: 7890

     Primary database: <not set>
     New primary database:  <not set>
```

Example 2.

```
ASGCTL> ASGCTL> show env

     ASG Server Connection:
        Host: standbyinfra
        Port: 7890

Gathering information from the database orcl

     Primary database: :
        User: sys
        Service: orcl
        Role: The database role is
               PHYSICAL STANDBY


     New primary database:  <not set>
```

show operation

Shows all operations on all nodes of the topology to which the OracleAS Guard client is connected for the current session.

Format

show op[eration] [full] [[his]tory]

Parameters

full
For all operations, shows the operation number, the job name, the job owner's user name, the job ID, the time the operation began, the time the operation ended, the elapsed time for the operation, and all tasks belonging to this job.

history
For only operations that are not running, shows the operation number and the job name.

Usage Notes

None.

Example

The following examples show the status of the current operation.

```
ASGCTL> show operation
**********************************
OPERATION: 19
  Status: running
  Elapsed Time: 0 days, 0 hours, 0 minutes, 28 secs
  TASK: syncFarm
    TASK: backupFarm
      TASK: fileCopyRemote
      TASK: fileCopyRemote
    TASK: restoreFarm
      TASK: fileCopyLocal
```

The following example shows the history of all operations.

```
ASGCTL> show op his
**********************************
OPERATION: 7
  Status: success
  Elapsed Time: 0 days, 0 hours, 0 minutes, 0 secs
  TASK: getTopology
    TASK: getInstance
**********************************
OPERATION: 16
  Status: success
  Elapsed Time: 0
 days, 0 hours, 0 minutes, 0 secs
  TASK: getTopology
    TASK: getInstance
**********************************
OPERATION: 19
```

```
Status: success
Elapsed Time: 0 days, 0 hours, 1 minutes, 55 secs
TASK: syncFarm
  TASK: backupFarm
    TASK: fileCopyRemote
    TASK: fileCopyRemote
  TASK: restoreFarm
    TASK: fileCopyLocall
```

shutdown

Shuts down a running OracleAS Guard server to which the OracleAS Guard client is connected. Use this command only on a host system where OPMN is not running and you are following the procedure to clone an instance or clone a topology.

Format

shutdown [local]

Parameters

local
When specified shuts down the OracleAS Guard server of the local Oracle home of asgctl.

Usage Notes

The OracleAS Guard server must have been started using the asgctl startup command and not the OPMN opmnctl command startproc.

Example

The following example shuts down the OracleAS Guard server on a host system in which OPMN is not running.

```
> asgctl.sh shutdown
```

shutdown topology

Shuts down the OracleAS component services across the topology, while OracleAS Guard server and OPMN will continue to run.

Format

shutdown topology

Parameters

None.

Usage Notes

This is a convenient command for shutting down the entire topology. Use the startup topology command to start it up again.

This command will shutdown OracleAS services such as OID, OC4J, WebCache, and so forth.

Example

The following example shuts down the prodinfra production topology.

```
ASGCTL> shutdown topology
Generating default policy for this operation

prodinfra:7890
    Shutting down each instance in the topology

asmid2:7890 (home /private1/OraHome2/asmid2)
    Shutting down component HTTP_Server
    Shutting down component OC4J
    Shutting down component dcm-daemon
    Shutting down component LogLoader

asmid1:7890 (home /private1/OraHome/asmid1)
    Shutting down component HTTP_Server
    Shutting down component OC4J
    Shutting down component dcm-daemon
    Shutting down component LogLoader

prodinfra:7890 (home /private1/OraHome2/asr1012)
    Shutting down component OID
    Shutting down component HTTP_Server
    Shutting down component OC4J
    Shutting down component dcm-daemon
    Shutting down component LogLoader
ASGCTL>
```

startup

Starts up an OracleAS Guard server from the asgctl prompt. Use this command only on a host system where OPMN is not running and you are following the procedure to clone an instance or clone a topology.

Format

startup

Parameters

None.

Usage Notes

None.

Example

The following example shuts down the OracleAS Guard server on a host system in which OPMN is not running.

```
> asgctl.sh startup
```

startup topology

Starts up a shutdown topology by starting up the OracleAS component services across the topology.

Format

startup topology

Parameters

none

Usage Notes

This is a convenient command for starting up the entire topology after it was shut down using the shutdown topology command.

This command will start up OracleAS services such as OID, OC4J, WebCache, and so forth. The startup topology command will perform the equivalent of an opmnctl startup command across each instance of the topology.

Example

The following example starts up the production topology.

```
ASGCTL> startup topology
Generating default policy for this operation

profinfra:7890
    Starting each instance in the topology

prodinfra:7890 (home /private1/OraHome2/asr1012)
    Executing opmnctl startall command

asmid1:7890 (home /private1/OraHome/asmid1)
    Executing opmnctl startall command

asmid2:7890 (home /private1/OraHome2/asmid2)
    Executing opmnctl startall command
ASGCTL>
```

stop operation

Stops a specific operation that is running on the server.

Format

stop op[eration] <op #>

Parameters

op #
The number of the operation.

Usage Notes

The number of the operation that is running on the server can be determined from a show operation command.

Example

The following example first shows the running operation (15) on the server and then the stop operation command stops this operation.

```
ASGCTL> show operation
**********************************
OPERATION: 15
  Status: running
  Elapsed Time: 0 days, 0 hours, 1 minutes, 35 secs
  TASK: instantiateFarm
    TASK: verifyFarm

ASGCTL> stop operation 15
```

switchover topology

During a scheduled outage of the production site, performs the switchover operation from the production site to the standby site.

Format

switchover topology to <standby_topology_host>[:<port>] [using policy <file>]

Parameters

standby_topology_host
Name of the standby host system. This parameter is required because it directs the coordinating OracleAS Guard server instance to discover the instances that make up the standby site. This host system must be a member of the standby topology.

port
The port number of the standby host system for the OracleAS Guard server in its Oracle home.

using policy <file>
Full path and file specification for the XML policy file.

Usage Notes

On the primary infrastructure system, make sure the emagent process is stopped. Otherwise, you may run into the following error when doing a switchover operation because the emagent process has a connection to the database:

```
prodinfra: -->ASG_DGA-13051: Error performing a physical standby switchover.
prodinfra: -->ASG_DGA-13052: The primary database is not in the proper state to perform a switchover.  State is
"SESSIONS ACTIVE"
```

On UNIX systems, to stop the emagent process, stop the Application Server Control, which is called iasconsole, as follows:

```
> <ORACLE_HOME>/bin/emctl stop iasconsole
```

On UNIX systems, to check to see if there is an emagent process running, do the following:

```
> ps -ef | grep emagent
```

On UNIX systems, if after performing the stop iasconsole operation, the emagent process is still running, get its process ID (PID) as determined from the previous ps command and stop it as follows:

```
> kill -9 <emagent-pid>
```

On Windows systems, open the Services control panel. Locate the OracleAS10gASControl service and stop this service.

Make sure OracleAS Infrastructure database is running on the primary topology before performing a switchover operation. Also, the OracleAS Infrastructure database information must be set by using the set primary database asgctl command.

The global DNS names are used to direct the switchover. This will be different than the HA naming utilized in the OracleAS Disaster Recovery environment. The discovery

mechanism automatically maps the topology to the corresponding peer, based off local name resolution.

As part of the OracleAS Guard switchover operation, an implicit sync topology operation is performed to make sure the topologies are identical. In addition OPMN automatically starts the OracleAS Guard server on the "new" standby Infrastructure node and this server will run indefinitely, and in turn, starts the OracleAS Guard server on the other nodes in the "new" standby topology and each of these is a transient server.

For the switchover policy file, by default the success requirement attribute is set to optional for all instances (middle tier and OracleAS Metadata Repository) and mandatory for the Oracle Internet Directory home.

During a switchover operation, the `opmn.xml` file is copied from the primary site to the standby site. For this reason, the value of the TMP variable must be defined the same in the `opmn.xml` file on both the primary and standby sites, otherwise this switchover operation will fail with a message that it could not find a directory. Therefore, make sure the TMP variable is defined identically and resolves to the same directory structure on both sites before attempting a switchover operation.

When performing a switchover operation from a primary site with two Oracle Identity Management instances running (im.machineA.us.oracle.com and im.machineB.us.oracle.com) to a standby site representing an asymmetric topology with only one Oracle Identity Management instance running (im.machineA.us.oracle.com), meaning that the other node (im.machineB.us.oracle.com) is to be ignored on the switchover site, the system administrator must not only edit the `switchover_policy.xml` policy file to indicate that this other node is to be set to Ignore, but the system administrator must also shut down all processes running on that node (im.machineB.us.oracle.com) in order for the switchover operation to be successful.

When performing a switchover operation from a primary site with two middle tiers, for example core1 and core2 instances registered in the Oracle Internet Directory, to a standby site representing an asymmetric topology with only one middle tier core1, the standby site actually has both core1 and core2 middle tiers registered in the Oracle Internet Directory. The `switchover_policy.xml` policy file is edited to ignore the core2 middle tier that does not exist on the standby site during the switchover operation. However, it should be noted that the Oracle Internet Directory, which is stored in an Oracle database, is identical for both the production site topology and the standby site topology and therefore a core2 middle tier is also shown to be registered in the Oracle Internet Directory on the standby site topology. For this reason, you cannot install to that standby site topology the same core2 middle tier with the hope of making this into a symmetric topology again. This is a strict limitation for switchover operations using asymmetric standby topologies.

When the discover topology command is issued following a switchover operation and the asymmetric standby site topology originally had one or more fewer middle tiers (for example, instA and instB) than there were in the original production site topology (instA, instB, and instC), a warning error message displays for each missing instance of a middle tier (instC, in this case). This warning error message is expected and can be ignored. When a discover topology to command is issued following a switchover operation, OracleAS Server Guard reads the Oracle Internet Directory information, which is an exact copy of the original primary site Oracle Internet Directory information on this new primary site (former standby site). Because this Oracle Internet Directory information is identical to the original primary site Oracle Internet Directory information, when OracleAS Server Guard visits the host/home of each

instance of these middle tiers to verify their existence, it finds that some do not exist, and issues the warning.

See Section 12.1, "Information Common to OracleAS Guard asgctl Commands" and Section 12.2, "Information Specific to a Small Set of OracleAS Guard Commands" for more information.

Example

The following example performs a switchover operation to a standby site known by DNS as standbyinfra.

```
ASGCTL> connect asg prodinfra ias_admin/adminpwd
Successfully connected to prodinfra:7890
ASGCTL> set primary database sys/testpwd@asdb
ASGCTL> switchover topology to standbyinfra
Generating default policy for this operation
prodinfra:7890
    Switchover each instance in the topology to standby topology
prodinfra:7890 (home /private1/OraHome2/asr1012)
    Connecting to the primary database asdb.us.oracle.com
    Gathering information from the primary database asdb.us.oracle.com
    Shutting down each instance in the topology
.
.
.
prodinfra:7890
     HA directory exists for instance asr1012.infra.us.oracle.com
asmid2:7890
     HA directory exists for instance asmid2.asmid2.us.oracle.com
asmid1:7890
     HA directory exists for instance asmid1.asmid1.us.oracle.com
standbyinfra:7890
     HA directory exists for instance asr1012.infra.us.oracle.com
asmid2:7890
     HA directory exists for instance asmid2.asmid2.us.oracle.com
asmid1:7890
     HA directory exists for instance asmid1.asmid1.us.oracle.com
prodinfra:7890
    Verifying that the topology is symmetrical in both primary and standby configuration
ASGCTL>

# Command to use if you are using a policy file
# switchover topology to standbyinfra using policy <file>
```

sync topology

Synchronizes the standby site with the primary site to ensure that the two sites are consistent. The sync topology operation applies database redo logs for OracleAS Infrastructures to the standby site in conjunction with synchronizing external configuration files across the topology.

Format

sync topology to <standby_topology_host>[:<port>] [full | incr[emental]] [using policy <file>]

Parameters

standby_topology_host
Name of the standby site host system. This parameter is required because it directs the coordinating OracleAS Guard server instance to discover the instances that make up the standby site. This host system must be a member of the standby topology.

port
The port number of the standby host system for the OracleAS Guard server in its Oracle home.

full | incremental
The synchronization of the standby site with the primary site to make the standby site consistent can be either "full" or "incremental". The default is "incremental". By default, if a full backup has not been performed, an incremental backup operation will not be performed. Instead, a full backup operation will be performed.

using policy file
Full path and file specification for the XML policy file.

Usage Notes

By default an incremental synchronization is performed to make the standby site consistent with the primary site, which offers the best performance. However, there may be three circumstances when specifying a full synchronization should be used.

- When you want to force a full synchronization to happen, such as synchronizing the standby site completely at a specific point in time (currently) with the primary site.
- When you know there are many transactional changes over a short period of time on the primary site that must be synchronized with the secondary site.
- When you know that there are a large accumulation of transactional changes over a long period of time on the primary site that must be synchronized with the secondary site.

The sync operation performs an implicit verify operation.

For the sync policy file, by default the success requirement attribute is set to mandatory for all instances.

See Section 12.1, "Information Common to OracleAS Guard asgctl Commands" and Section 12.2, "Information Specific to a Small Set of OracleAS Guard Commands" for more information.

Example

The following example synchronizes the specified standby site with the coordinating OracleAS Guard server (the primary site). By default the sync mode is incremental.

```
ASGCTL> connect asg prodinfra ias_admin/adminpwd
Successfully connected to prodinfra:7890
ASGCTL> set primary database sys/testpwd@asdb
Checking connection to database asdb
ASGCTL> sync topology to standbyinfra
Generating default policy for this operation
prodinfra:7890
    Synchronizing each instance in the topology to standby topology
prodinfra:7890 (home /private1/OraHome2/asr1012)
    Starting backup of topology ""
       Backing up and copying data to the standby topology
    Backing up each instance in the topology
    Starting backup of instance "asr1012.infra.us.oracle.com"
    Configuring the backup script
asmid1:7890 (home /private1/OraHome/asmid1)
    Starting backup of instance "asmid1.asmid1.us.oracle.com"
asmid2:7891 (home /private1/OraHome/asmid2)
    Starting backup of instance "asmid2.asmid2.us.oracle.com"
.
.
.
asmid2:7890 (home /private1/OraHome2/asr1012)
    Starting backup/synchronization of database "asdb.us.oracle.com"
    Starting restore/synchronization of database "asdb.us.oracle.com"
    Synchronizing topology completed successfully
ASGCTL>

# Command to use if you are using a policy file
# sync topology to standbyinfra using policy <file>
```

verify topology

Validates that the primary topology is running and the configuration is valid. If a standby topology is specified, compares the primary topology to which the local host system is a member with the standby topology to validate that they are consistent with one another and conform to the requirements for OracleAS Disaster Recovery.

Format

verify topology [with <host>[:<port>]] [using policy <file>]

Parameters

host
Name of the standby host system. This host system must be a member of the standby topology.

port
The port number of the host system for the OracleAS Guard server in its Oracle home.

using policy <file>
Full path and file specification for the XML policy file.

Usage Notes

If the host system name is not specified, the topology in which the local host system participates will be verified for local OracleAS Disaster Recovery rules.

If the standby host system name is specified, the topology at the standby site will be verified along with the production topology for both local rules and distributed OracleAS Disaster Recovery rules, and the symmetry between the primary and standby sites is also checked.

For the verify policy file, by default the success requirement attribute is set to optional for all OracleAS homes (middle tier and OracleAS Metadata Repository) and mandatory for the Oracle Internet Directory home.

See Section 12.1, "Information Common to OracleAS Guard asgctl Commands" and Section 12.2, "Information Specific to a Small Set of OracleAS Guard Commands" for more information.

Example

The following example validates that the primary topology is running and the configuration is valid.

```
ASGCTL> connect asg ias_admin/iastest2
Successfully connected to prodinfra:7890
ASGCTL> verify topology
Generating default policy for this operation
prodinfra:7890
     HA directory exists for instance asr1012.infra.us.oracle.com
asmid2:7890
     HA directory exists for instance asmid2.asmid2.us.oracle.com
asmid1:7890
     HA directory exists for instance asmid1.asmid1.us.oracle.com
ASGCTL>
```

The following example validates that the topology to which the local host system is a member is consistent with the standby topology to which the host system standbyinfra is a member.

```
ASGCTL> connect asg prodinfra ias_admin/adminpwd
Successfully connected to prodinfra:7890
ASGCTL> set primary database sys/testpwd@asdb
Checking connection to database asdb
ASGCTL> verify topology with standbyinfra
Generating default policy for this operation
prodinfra:7890
     HA directory exists for instance asr1012.infra.us.oracle.com
asmid2:7890
     HA directory exists for instance asmid2.asmid2.us.oracle.com
asmid1:7890
     HA directory exists for instance asmid1.asmid1.us.oracle.com
standbyinfra:7890
     HA directory exists for instance asr1012.infra.us.oracle.com
asmid2:7890
     HA directory exists for instance asmid2.asmid2.us.oracle.com
asmid1:7890
     HA directory exists for instance asmid1.asmid1.us.oracle.com
prodinfra:7890
    Verifying that the topology is symmetrical in both primary and standby configuration
ASGCTL>

# Command to use if you are using a policy file
# verify topology using policy <file>
```

dump farm (Deprecated)

Directs asgctl to write detailed information about the farm to the specified file.

> **Note:** The dump farm command is deprecated beginning with OracleAS release 10.1.2.0.2. Use the dump topology command, which supports the OracleAS Disaster Recovery topology concept in current and future OracleAS releases.

Format

dump farm [to <file>]

Parameters

to <file>
Name of file on the OracleAS Guard client node where the detailed output is to be written.

Usage Notes

None.

Example

See the dump topology command for an example.

instantiate farm (Deprecated)

Instantiates a farm to a standby site by discovering the current farm definition at the production and standby sites, verifying that each complies with the OracleAS Disaster Recovery rules and restrictions of the current OracleAS software deployed on these systems prior to creation. Also synchronizes the standby site with the primary site so that the primary and standby sites are consistent.

> **Note:** The instantiate farm to command is deprecated beginning with OracleAS release 10.1.2.0.2. Use the instantiate topology command, which supports the OracleAS Disaster Recovery topology concept in current and future OracleAS releases.

Format

```
instantiate farm to <standby_farm_host>[:<port>]
```

Parameters

standby_farm_host
Name of the standby host system. This parameter is required because it directs the coordinating OracleAS Guard server instance to discover the instances that make up the standby site. This host system must be a member of the standby farm.

port
The port number of the OracleAS Guard server in its Oracle home.

Usage Notes

The production local system must be part of an Oracle Notification Server (ONS) farm for the site.

The standby host must be part of an ONS farm for the standby site and must be symmetrical to the farm of the production farm.

Make sure OracleAS Infrastructure database is running on the primary farm before performing an instantiating farm operation. Also, the OracleAS Infrastructure database information must be set by using the set primary database asgctl command.

The global DNS names are used to direct the instantiation. This will be different than the HA naming utilized in the OracleAS Disaster Recovery environment. The discovery mechanism automatically maps the farm to the corresponding peer, based off local name resolution.

Example

See the instantiate topology command for an example.

shutdown farm (Deprecated)

Shuts down a running farm.

> **Note:** The shutdown farm command is deprecated beginning with OracleAS release 10.1.2.0.2. Use the shutdown topology command, which supports the OracleAS Disaster Recovery topology concept in current and future OracleAS releases.

Format

shutdown farm

Parameters

None.

Usage Notes

This is a convenient command for shutting down the entire farm. Use the startup farm command to start it up again.

Example

See the shutdown topology command for an example.

startup farm (Deprecated)

Starts up a shutdown farm.

> **Note:** The startup farm command is deprecated beginning with OracleAS release 10.1.2.0.2. Use the startup topology command, which supports the OracleAS Disaster Recovery topology concept in current and future OracleAS releases.

Format

startup farm

Parameters

None

Usage Notes

This is a convenient command for starting up the entire farm after it was shut down using the shutdown farm command.

Example

See the startup topology command for an example.

switchover farm (Deprecated)

During a scheduled outage of the production site, performs the switchover operation from the production site to the standby site.

> **Note:** The switchover farm to command is deprecated beginning with OracleAS release 10.1.2.0.2. Use the switchover topology command, which supports the OracleAS Disaster Recovery topology concept in current and future OracleAS releases.

Format

switchover farm to <standby_farm_host>[:<port>]

Parameters

standby_farm_host
Name of the farm host system. This parameter is required because it directs the coordinating OracleAS Guard server instance to discover the instances that make up the standby site. This host system must be a member of the standby farm.

port
The port number of the standby host system for the OracleAS Guard server in its Oracle home.

Usage Notes

On the primary Infrastructure system, make sure the emagent process is stopped. Otherwise, you may run into the following error when doing a switchover operation because the emagent process has a connection to the database:

```
prodinfra: -->ASG_DGA-13051: Error performing a physical standby switchover.
prodinfra: -->ASG_DGA-13052: The primary database is not in the proper state to perform a switchover.  State is
"SESSIONS ACTIVE"
```

On UNIX systems, to stop the emagent process, stop the Application Server Control, which is called iasconsole, as follows:

```
> <ORACLE_HOME>/bin/emctl stop iasconsole
```

On UNIX systems, to check to see if there is an emagent process running, do the following:

```
> ps -ef | grep emagent
```

On UNIX systems, if after performing the stop iasconsole operation, the emagent process is still running, get its process ID (PID) as determined from the previous ps command and stop it as follows:

```
> kill -9 <emagent-pid>
```

On Windows systems, open the Services control panel. Locate the OracleAS10gASControl service and stop this service.

The production local system must be part of an Oracle Notification Server (ONS) farm for the site.

The standby host must be part of an ONS farm for the standby site and must be symmetrical to the farm of the production farm.

Make sure OracleAS Infrastructure database is running on the primary farm before performing a switchover operation. Also, the OracleAS Infrastructure database information must be set by using the set primary database asgctl command.

The global DNS names are used to direct the switchover. This will be different than the HA naming utilized in the OracleAS Disaster Recovery environment. The discovery mechanism automatically maps the farm to the corresponding peer, based off local name resolution.

As part of the OracleAS Guard switchover operation, an implicit sync farm operation is performed to make sure the farms are identical. In addition, OPMN automatically starts the OracleAS Guard server on the "new" standby Infrastructure node and this server will run indefinitely. In turn, it starts the OracleAS Guard server on the other nodes in the "new" standby farm and each of these is a transient server.

Example

See the switchover topology command for an example.

sync farm (Deprecated)

Synchronizes the standby site with the primary site to ensure that the two sites are consistent. The sync topology operation applies database redo logs for OracleAS Infrastructures to the standby site in conjunction with synchronizing external configuration files across the topology.

> **Note:** The sync farm to command is deprecated beginning with OracleAS release 10.1.2.0.2. Use the sync topology command, which supports the OracleAS Disaster Recovery topology concept in current and future OracleAS releases.

Format

sync farm to <standby_farm_host>[:<port>] [full | incr[emental]]

Parameters

standby_farm_host
Name of the standby site host system. This parameter is required because it directs the coordinating OracleAS Guard server instance to discover the instances that make up the standby site. This host system must be a member of the standby farm.

port
The port number of the standby host system for the OracleAS Guard server in its Oracle home.

full | incremental
The synchronization of the standby site with the primary site to make the standby site consistent can be either "full" or "incremental". The default is "incremental". By default, if a full backup has not been performed, an incremental backup operation will not be performed. Instead, a full backup operation will be performed.

Usage Notes

By default `sync_mode` is incremental and offers the best performance. However, there may be three circumstances when specifying a `sync_mode` of full should be used.

- When you want to force a full synchronization to happen, such as synchronizing the standby site completely at a specific point in time (currently) with the primary site.
- When you know there are many transactional changes over a short period of time on the primary site that must be synchronized with the secondary site.
- When you know that there is a large accumulation of transactional changes over a long period of time on the primary site that must be synchronized with the secondary site.

Example

See the sync topology command for an example.

verify farm (Deprecated)

Validates that the primary farm is running and the configuration is valid. If a standby farm is specified, compares the primary farm to which the local host system is a member with the standby farm to validate that they are consistent with one another and conform to the requirements for OracleAS Disaster Recovery.

> **Note:** The verify farm command is deprecated beginning with OracleAS release 10.1.2.0.2. Use the verify topology command, which supports the OracleAS Disaster Recovery topology concept in current and future OracleAS releases.

Format

verify farm [with <host>[:<port>]]

Parameters

host
Name of the standby host system. This host system must be a member of the standby farm.

port
The port number of the OracleAS Guard server in its Oracle home.

Usage Notes

If the host system name is not specified, the farm in which the local host system participates will be verified for local OracleAS Disaster Recovery rules.

If the standby host system name is specified, the farm at the standby site will be verified along with the production farm for both local rules and distributed OracleAS Disaster Recovery rules, and the symmetry between the primary and standby sites is also checked.

Example

See the verify topology command for an examples.

13

Manual Sync Operations

The following manual sync operations must be performed if for some reason the secondary (standby) site is not synchronized with the primary site and you are performing regular backup operations of the primary site middle tier and OracleAS Infrastructure configuration files as described in Section 13.1.1, "Manually Backing Up the Production Site". Then you will need to restore the backup configuration files as described in Section 13.1.2, "Manually Restoring to Standby Site". After restoring the configuration files (OracleAS Infrastructure and Middle Tier) on the standby site, then proceed to Step 2 as described in "Site Failover Operations" on page 11-43.

13.1 Manually Synchronizing Baseline Installation with Standby Site Without Using OracleAS Guard asgctl Command-line Utility

> **Note:** This section and Section 13.1.1, "Manually Backing Up the Production Site" and Section 13.1.2, "Manually Restoring to Standby Site" are retained here for the special case where the standby site is not synchronized with the primary site. In this case, on the standby site, you must restore the most recently backed up configuration files as described in Section 13.1.2, "Manually Restoring to Standby Site".
>
> If you are using asgctl to continually synchronize the secondary (standby) site with the primary site, then both sites should already be synchronized and you do not need to manually perform a restore operation and you can begin with Step 2 in "Site Failover Operations" on page 11-43 to recover from an unplanned outage.

Once Oracle Data Guard has been set up between the production and standby sites, the procedure for synchronizing the two sites can be carried out. An initial synchronization should be done, before the production site is used, in order to obtain a baseline snapshot of the post-installation production site onto the standby site. This baseline can then be used to recover the production site configuration on the standby site if needed later.

In order to obtain a consistent point-in-time snapshot of the production site, the information stored in the OracleAS Infrastructure database and the Oracle Application Server-related configuration files in the middle-tier and OracleAS Infrastructure hosts must be synchronized at the same time. Synchronization of the configuration files can be done by backing up the files and restoring them on the standby hosts using the Oracle Application Server Backup and Recovery Tool. For the OracleAS Infrastructure database, synchronization is done using Oracle Data Guard by shipping the archive

logs to the standby OracleAS Infrastructure and applying these logs in coordination with the restoration of the configuration files.

The sequence of steps for the baseline synchronization (which can also be used for future synchronizations) are:

- Shipping OracleAS Infrastructure Database Archive Logs
- Backing Up Configuration Files (OracleAS Infrastructure and Middle Tier)
- Restoring Configuration Files (OracleAS Infrastructure and Middle Tier)
- Restoring the OracleAS Infrastructure Database - Applying Log Files

These steps are detailed in the following two main sections.

13.1.1 Manually Backing Up the Production Site

The main strategy and approach to synchronizing configuration information between the production and standby sites is to synchronize the backup of OracleAS Infrastructure and middle-tier configuration files with the application of log information on the standby OracleAS Infrastructure database.

For Oracle Application Server, not all the configuration information is in the OracleAS Infrastructure database. The backup of the database files needs to be kept synchronized with the backup of the middle-tier and OracleAS Infrastructure configuration files. Due to this, log-apply services should not be enabled on the standby database. The log files from the production OracleAS Infrastructure are shipped to the standby OracleAS Infrastructure but are not applied.

The backup process of the production site involves backing up the configuration files in the middle-tier and OracleAS Infrastructure nodes. Additionally, the archive logs for the OracleAS Infrastructure database are shipped to the standby site.

The procedures to perform the backups and the log ship are discussed in the following sections:

- Shipping OracleAS Infrastructure Database Archive Logs
- Backing Up Configuration Files (OracleAS Infrastructure and Middle Tier)

IMPORTANT: Ensure that no configuration changes are going to be made to the Oracle Application Server system (underlying configuration files and OracleAS Infrastructure database) as you perform the steps in this section.

Note: At the minimum, the backup and restoration steps discussed in this section and the "Manually Restoring to Standby Site" section should be performed whenever there is any administration change in the production site (inclusive of changes to the OracleAS Infrastructure database and configuration files on the middle-tier and OracleAS Infrastructure nodes). On top of that, scheduled regular backups and restorations should also be done (for example, on a daily or twice weekly basis). See the *Oracle Application Server Administrator's Guide* for more backup and restore procedures.

13.1.1.1 Shipping OracleAS Infrastructure Database Archive Logs

After installing the OracleAS Disaster Recovery solution, Oracle Data Guard should have been installed in both the production and standby databases. The steps for shipping the archive logs from the production OracleAS Infrastructure database to the standby OracleAS Infrastructure database involve configuring Oracle Data Guard and executing several commands for both the production and standby databases. Execute the following steps to ship the logs for the OracleAS Infrastructure database:

1. If not disabled already, disable log-apply services by running the following SQLPLUS statement on the standby host:

   ```
   SQL> alter database recover managed standby database cancel;
   ```

2. Run the following command to perform a log switch on the production OracleAS Infrastructure database. This ensures that the latest log file is shipped to the standby OracleAS Infrastructure database

   ```
   SQL> alter system switch logfile;
   ```

3. In normal operation of the production site, the production database frequently ships log files to the standby database but are not applied. At the standby site, you want to apply the logs that are consistent up to the same time that the production site's configuration files are backed up. The following SQL statement encapsulates all OracleAS Infrastructure database changes into the latest log and allows the Oracle Data Guard transport services to transport this log to the OracleAS Infrastructure in the standby site:

   ```
   SQL> select first_change# from v$log where status='CURRENT';
   ```

   A SCN or sequence number is returned, which essentially represents the timestamp of the transported log.

4. Note down the SCN number as you will need this for the restoration of the production database changes on the standby site.

Continue to the next section to back up the configuration files on the middle-tier host(s) and OracleAS Infrastructure host.

13.1.1.2 Backing Up Configuration Files (OracleAS Infrastructure and Middle Tier)

Use the instructions in this section to back up the configuration files. The instructions require the use of the OracleAS Backup and Recovery Tool. They assume you have installed and configured the tool on each OracleAS installation (middle tier and OracleAS Infrastructure) as it needs to be customized for each installation. Refer to *Oracle Application Server Administrator's Guide* for more details about that tool, including installation and configuration instructions.

For each middle-tier and OracleAS Infrastructure installation, perform the following steps (the same instructions can be used for the middle-tier and OracleAS Infrastructure configuration files):

1. After performing the installation and configuration steps detailed in the *Oracle Application Server Administrator's Guide*, for the Oracle Application Server Backup and Recovery Tool, the variables `oracle_home`, `log_path`, and `config_backup_path` in the tool's configuration file, `config.inp`, should have the appropriate values. Also, the following command for the tool should have been run to complete the configuration:

   ```
   perl bkp_restore.pl -m configure_nodb
   ```

In Windows, the Perl executable can be found in *<ORACLE_HOME>*\perl*<perl_version>*\bin\MSWin32-x86.

If you have not completed these tasks, do so before continuing with the ensuing steps.

2. Execute the following command to back up the configuration files from the current installation:

```
perl bkp_restore.pl -v -m backup_config
```

This command creates a directory in the location specified by the config_backup_path variable specified in the config.inp file. The directory name includes the time of the backup. For example: config_bkp_2003-09-10_13-21.

3. A log of the backup is also generated in the location specified by the log_path variable in the config.inp file. Check the log files for any errors that may have occurred during the backup process.
4. Copy the OracleAS Backup and Recovery Tool's directory structure and contents from the current node to its equivalent in the standby site. Ensure that the path structure on the standby node is identical to that on the current node.
5. Copy the backup directory (as defined by config_backup_path) from the current node to its equivalent in the standby site. Ensure that the path structure on the standby node is identical to that on the current node.
6. Repeat the steps above for each Oracle Application Server installation in the production site (middle tier and OracleAS Infrastructure).

Note: There are two important items that should be maintained consistently between the production and standby sites. The directory names should be the same and the correlation of SCN to a given backup directory should be noted at both sites in administration procedures.

13.1.2 Manually Restoring to Standby Site

After backing up the configuration files from the middle-tier Oracle Application Server instances and OracleAS Infrastructure together with the OracleAS Infrastructure database, restore the files and database in the standby site using the instructions in this section, which consists of the following sub-sections:

- Restoring Configuration Files (OracleAS Infrastructure and Middle Tier)
- Restoring the OracleAS Infrastructure Database - Applying Log Files

13.1.2.1 Restoring Configuration Files (OracleAS Infrastructure and Middle Tier)

Restoring the backed up files from the production site requires the OracleAS Backup and Recovery Tool that was used for the backup. The instructions in this section assume you have installed and configured the tool on each OracleAS installation in the standby site, both in the middle-tier and OracleAS Infrastructure nodes. Refer to *Oracle Application Server Administrator's Guide* for instructions on how to install the tool.

For each middle-tier and OracleAS Infrastructure installation in the standby site, perform the following steps (the same instructions can be used for the middle-tier and OracleAS Infrastructure configuration files):

1. Check that the OracleAS Backup and Recovery Tool's directory structure and the backup directory from the equivalent installation in the production site are present in the current node.
2. Stop the Oracle Application Server instances and their processes so that no modification of configuration files can occur during the restoration process. Use the following OPMN command:

   In UNIX:

   ```
   <ORACLE_HOME>/opmn/bin/opmnctl stopall
   ```

   In Windows:

   ```
   <ORACLE_HOME>\opmn\bin\opmnctl stopall
   ```

   Check that all relevant processes are no longer running. In UNIX, use the following command:

   ```
   ps -ef | grep <ORACLE_HOME>
   ```

   In Windows, press `<ctrl><alt><del>` to bring up the Task Manager and verify that the processes have stopped.
3. Configure the backup utility for the Oracle home.

   This can be accomplished either by configuring the OracleAS Backup and Recovery Tool for the Oracle home or copying the backup configuration file, `config.inp`, from the production site peer. Below is an example of running the OracleAS Backup and Recovery Tool configuration option:

   ```
   perl bkp_restore.pl -v -m configure_nodb
   ```

   In Windows, the Perl executable can be found in *<ORACLE_HOME>*`\perl\`*<perl_version>*`\bin\MSWin32-x86`.
4. Execute the following command to view a listing of the valid configuration backup locations:

   ```
   perl bkp_restore.pl -v -m restore_config
   ```

5. Restore the configuration files using the following command:

   ```
   perl bkp_restore.pl -v -m restore_config -t <backup_directory>
   ```

   where <*backup_directory*> is the name of the directory with the backup files that was copied from the production site. For example, this could be `config_bkp_2003-09-10_13-21`.
6. Check the log file specified in `config.inp` for any errors that may have occurred during the restoration process.
7. Repeat the steps above for each Oracle Application Server installation in the production site (middle tier and OracleAS Infrastructure).

13.1.2.2 Restoring the OracleAS Infrastructure Database - Applying Log Files

During the backup phase, you executed several instructions to ship the database log files from the production site to the standby site up to the SCN number that you recorded as per instructed. To restore the standby database to that SCN number, apply the log files to the standby OracleAS Infrastructure database using the following SQLPLUS statement:

```
SQL> alter database recover automatic from '/private/oracle/oracleas/standby/' standby
```

```
database until change <SCN>;
```

(In Windows, substitute the path shown above appropriately.)

With this command executed and the instructions to restore the configuration files completed on each middle-tier and OracleAS Infrastructure installation, the standby site is now synchronized with the production site. However, there are two common problems that can occur during the application of the log files: errors caused by the incorrect specification of the path and gaps in the log files that have been transported to the standby site.

The following are methods of resolving these problems:

1. Find the correct log path.

   On the standby OracleAS Infrastructure database, try to determine location and number of received archive logs using the following SQLPLUS statement:

   ```
   SQL> show parameter standby_archive_dest

   NAME                      TYPE        VALUE
   ------------------------- ----------- ---------------------------------
   standby_archive_dest      string      /private/oracle/oracleas/standby/
   ```

   (The previous example shows the UNIX path. The Windows equivalent path is shown in Windows systems.)

2. Use the log path obtained from the previous step to ensure that all log files have been transported.

   At the standby OracleAS Infrastructure database, perform the following:

   ```
   standby> cd /private/oracle/oracleas/standby
   standby> ls
   1_13.dbf  1_14.dbf  1_15.dbf  1_16.dbf  1_17.dbf  1_18.dbf  1_19.dbf
   ```

   (In Windows, use the command `cd` to change to the appropriate directory and dir to view the directory contents.)

   At the production OracleAS Infrastructure database, execute the following SQLPLUS statement:

   ```
   SQL> show parameter log_archive_dest_1

   NAME                 TYPE    VALUE
   -------------------- ------- ----------------------------------------------------
   log_archive_dest1   string  LOCATION=/private/oracle/oracleas/oradata MANDATORY
   log_archive_dest_10 string
   ```

   (The previous example shows the UNIX path. The Windows equivalent path is shown in Windows systems.)

3. Using the path specified in step 1, note the number and sequence of the log files. For example:

   ```
   production> cd /private/oracle/oracleas/oradata
   production> ls
   1_10.dbf  1_12.dbf  1_14.dbf  1_16.dbf  1_18.dbf  asdb
   1_11.dbf  1_13.dbf  1_15.dbf  1_17.dbf  1_19.dbf
   ```

   (In Windows, use the command `cd` to change to the appropriate directory and dir to view the directory contents.)

In the previous example, note the discrepancy where the standby OracleAS Infrastructure is missing files `1_10.dbf` through `1_12.dbf`. Since this gap in the log files happened in the past, it could be due to a problem with the historic setup involving the network used for the log transport. This problem has obviously been corrected and subsequent logs have been shipped. To correct the problem, copy (FTP) the log files to the corresponding directory on the standby OracleAS Infrastructure database host and re-attempt the SQLPLUS recovery statement shown earlier in this section.

14

OracleAS Disaster Recovery Site Upgrade Procedure

This chapter describes how to complete a full site Oracle Application Server Disaster Recovery (OracleAS Disaster Recovery) upgrade from OracleAS 10g (9.0.4) to OracleAS 10g (10.1.2.0.2). This procedure assumes that a successful release 9.0.4 to release 10.1.2 upgrade is possible for all the Oracle home types within the topology that define the site and extends these procedures in the OracleAS Disaster Recovery (DR) solution.

This site upgrades an existing supported DR implementation as documented in the Oracle Application Server Disaster Recovery chapter in *Oracle Application Server 10g High Availability Guide* for OracleAS 10g (9.0.4). This process will not upgrade a non-supported DR environment into an upgraded DR environment. Additionally, this procedure will utilize the standalone OracleAS Guard install within the existing release 9.0.4 Oracle homes and is worded using OracleAS Guard operational steps. If this environment is not possible, the equivalent manual steps can be performed, that is, the OracleAS Guard `sync topology` command equates to the site synchronization steps documented in the Oracle Application Server Disaster Recovery chapter in *Oracle Application Server 10g High Availability Guide* for OracleAS 10g (9.0.4).

14.1 Prerequisites

The following are prerequisites for performing a full DR site upgrade from OracleAS 10g (9.0.4) to OracleAS 10g (10.1.2.0.2):

- You must have a DR site configured according to the guidelines in Chapter 11, "OracleAS Disaster Recovery".
- The OracleAS Recovery Manager (formerly called OracleAS Backup/Restore utility) is installed in all Oracle homes of the both the production and standby sites. The Backup/Restore utility version to be used is the version that supports that release. For example, if you are performing a full Disaster Recovery site upgrade from Oracle 10g (9.0.4) to OracleAS 10g (10.1.2.0.2), then the Backup/Restore utility version must be OracleAS 10g (9.0.4).
- The OracleAS 10g (10.1.2.0.2) standalone install of OracleAS Guard, located on CDROM Disk 2, is installed in all OracleAS 10g (9.0.4) Oracle homes. See the OracleAS Disaster Recovery installation information in *Oracle Application Server Installation Guide* for more information.

14.2 Disaster Recovery Topology

The systems involved in this DR environment are contained in two sites, site A and site B. The initial roles of each are:

- Site A is the production site.
- Site B is the standby site.

Due to geographical separation of the sites, it is assumed that the current roles of each of these sites will be the final roles of these same sites at the end of this procedure. However, during the course of the procedure these roles do change. Thus, all references will be to the sites named A and B. Some of the terminology used may be confusing, depending on the role the site is maintaining at a particular point in time.

14.3 High-Level OracleAS Disaster Recovery Upgrade Steps

The following steps describe the OracleAS 10g (9.0.4) to OracleAS 10g (10.1.2.0.2) Disaster Recovery upgrade scenario. These steps refer to Infrastructure systems `infra1` and `infra2` on site A and site B, respectively.

1. Install the OracleAS 10g (10.1.2.0.2) standalone install of OracleAS Guard into each Oracle home on the production and standby sites.

   If multiple Oracle homes exist on the same system, ensure that different ports are configured for each of the OracleAS Guard servers in this configuration file. The default port number is 7890.

   ```
   <ORACLE_HOME>/dsa/dsa.conf
   ```

2. At the standby site [site B], start the OracleAS Guard server:

   ```
   <ORACLE_HOME>/opmn/bin/opmnctl startproc ias-component=DSA
   ```

3. At the production site [site A], connect to OracleAS Guard Infrastructure system `infra1` and perform a sync operation.

   This operation is used to ensure that the Oracle homes across the topology are logically synchronized.

   a. Invoke the asgctl client.

      ```
      On Unix systems
      <ORACLE_HOME>/dsa/bin/asgctl.sh

      On Windows systems
      <ORACLE_HOME>\dsa\bin\asgctl
      ```

   b. Perform a connect operation to site A Infrastructure system `infra1`.

      ```
      ASGCTL> connect asg infra1 ias_admin/<password>
      ```

   c. Set the primary database to the OracleAS Metadata Repository at site A.

      ```
      ASGCTL> set primary database sys/<password>@<site A's servicename>
      ```

   d. Discover the topology.

      ```
      ASGCTL> discover topology oidpassword=<oidpwd>
      ```

   e. Dump the topology.

      ```
      ASGCTL> dump topology to c:\policy_file_no_904_instances.txt
      ```

f. Edit the topology file, in this example named `policy_file_no_904_instances.txt` and set all the 9.0.4 instances to Ignore, for example:

```
<policy>
.
.
.
<instanceList successRequirement="Ignore">
   <instance>904Portal_3</instance>
</instanceList>
.
.
.
</policy>
```

Then use this policy file for all Disaster Recovery related operations.

g. Synchronize the standby site B Infrastructure system `infra2` with the production site using the edited policy file from Step 3f.

```
ASGCTL> sync topology to infra2 using policy c:\policy_file_no_904_
instances.txt
```

h. Ensure there are no changes to the environment through the duration of the upgrade procedure. Note that this does not mean changes to customer data, as this will be in a different database than the Identity Management (IM)/Metadata Repository (MR) data. However, in this model, the IM/MR data will not be able to be synchronized again during the upgrade procedure.

4. Connect to OracleAS Guard at the standby [site B] Infrastructure and failover.

The purpose of this step is to break the DR environment into two independent sites. This allows site B to be upgraded first. Once site B is upgraded, application level tests can be performed to ensure that the update was completed and that this site is operational. If you use this approach, then site A, production, is not really DR tolerant for the time period of the upgrade. Theoretically, another standby site could be established at this time as site B was upgraded.

The steps to follow to perform the OracleAS Guard failover operation are:

a. Perform a connect operation to site B Infrastructure system `infra2`.

```
ASGCTL> connect asg infra2 ias_admin/<password>
```

b. Set the new primary database to the OracleAS Metadata Repository at site B.

```
ASGCTL> set new primary database sys/<password>@<site B's servicename>
```

c. Perform the failover operation to this standby site, site B. The failover operation will start all the OPMN managed services across the topology equivalent of an opmnctl `startall` command.

```
ASGCTL> failover using policy c:\policy_file_no_904_instances.txt
```

5. Start the other services for the site at site B.

Any additional services needed for testing must be handled manually, such as applications, database jobs, Enterprise Manager, and so forth.

6. Perform an OracleAS upgrade to the site B systems [see *Oracle Application Server Upgrade and Compatibility Guide* for more information].

7. Test applications or note problems for resolution for the production site. Perform tests until you are satisfied the upgrade has been properly completed.
8. Redirect site access to site B, if desirable.
   a. During the next operation, site A will be upgraded, and Site B can provide some level of service during this upgrade procedure. Theoretically, all access can be given at this time. Once site B is upgraded, requests are serviced there, making this the production role of the DR environment. Once site A is upgraded, the software versions at both sites will be the same and a DR instantiate/sync operation will be possible (as performed in Step 12). If this approach is utilized, any updates made at the original production site [site A] will be lost.
   b. If Step 8a is implemented and site B becomes the production site, then ignore the restrictions in Step 3h because site A is about to be upgraded.
9. Perform an OracleAS upgrade to the site A systems [see *Oracle Application Server Upgrade and Compatibility Guide* for more information].
10. Test applications or note problems for resolution. Perform tests until you are satisfied the upgrade has been properly completed.

    At the end of this step, the two site upgrades are functionally equivalent and have been upgraded to OracleAS Disaster Recovery 10.1.2.0.2 Full site functionality has been enabled at site B and it is time to reestablish the production/standby relationship.
11. Stop the OracleAS Guard server in all the old OracleAS 9.0.4 Oracle homes, remove the `dsa.conf` file in the *<ORACLE_HOME>*`/dsa` directory on Unix systems or *<ORACLE_HOME>*`\dsa` directory on Windows systems, then restart the DSA process as well as the OPMN server on all the systems in the new OracleAS 10.1.2 Oracle homes.

    On UNIX systems:

    ```
    <ORACLE_HOME>/opmn/bin/opmnctl stopall
    <ORACLE_HOME>/opmn/bin/opmnctl startall
    <ORACLE_HOME>/opmn/bin/opmnctl startproc ias-component=DSA
    ```

    On Windows systems:

    ```
    <ORACLE_HOME>\opmn\bin\opmnctl stopall
    <ORACLE_HOME>\opmn\bin\opmnctl startall
    <ORACLE_HOME>\opmn\bin\opmnctl startproc ias-component=DSA
    ```

12. Use OracleAS Guard and perform a site instantiation from site B to site A if Step 8a is utilized.

    This step reestablishes the OracleAS Disaster Recovery environment between site B and Site A. In this sequence, site B is the production site, and site A is updated to mirror site B.

    Perform the following asgctl steps to complete this operation:

    a. Invoke the asgctl client.

    ```
    On Unix systems
    <ORACLE_HOME>/dsa/bin/asgctl.sh

    On Windows systems
    <ORACLE_HOME>\dsa\bin\asgctl
    ```

b. Perform a connect operation to site B's Infrastructure system `infra2`.

```
ASGCTL> connect asg infra2 ias_admin/<password>
```

c. Set the primary database to the OracleAS Metadata Repository at site B.

```
ASGCTL> set primary database sys/<password>@<site B's servicename>
```

d. Discover the topology.

```
ASGCTL> discover topology oidpassword=<oidpwd>
```

e. Dump the topology.

```
ASGCTL> dump topology to d:\policy_file_no_904_instances.txt
```

f. Edit the topology file, in this example named `policy_file_no_904_instances.txt` and set all the 9.0.4 instances to Ignore, for example:

```
<policy>
.
.
.
<instanceList successRequirement="Ignore">
   <instance>904Portal_3</instance>
</instanceList>
.
.
.
</policy>
```

Then use this policy file for all Disaster Recovery related operations.

g. Instantiate the topology to site A's standby Infrastructure system `infra1` using the edited policy file from Step 12f.

```
ASGCTL> instantiate topology to infra1 using policy d:\policy_file_no_904_
instances.txt
```

13. Perform a domain name system (DNS) switchover operation.

You would probably perform this step here to absorb the DNS timeout during the time period of the switchover operation. There will be end user access errors (service unavailable) until DNS, the site services, and the application have all been switched over and are running.

14. Use OracleAS Guard to perform a switchover operation from site B to site A.

The end goal is to have the same access at the end of upgrade as at the start of the process. Thus the roles have to be switched between the sites. Connect to the Infrastructure for site B, set the primary database, perform a discover topology, then perform a switchover to site A Infrastructure system `infra1` using the edited policy file from Step 12f.

```
ASGCTL> connect asg infra2 ias_admin/<password>
ASGCTL> set primary database sys/<password>@<site B's servicename>
ASGCTL> switchover topology to infra1 using policy d:\policy_file_no_904_
instances.txt
```

15. Note that an alternative to Steps 9 through 14 would be as follows:

a. Take down the production site [site A].

b. Perform an OracleAS upgrade to the site A systems [see *Oracle Application Server Upgrade and Compatibility Guide* for more information].

c. Perform site A to site B instantiation using OracleAS Guard.

16. Start or open up services at production site A for the application.

This completes the steps required for the OracleAS Disaster Recovery site upgrade procedure from OracleAS 10g (9.0.4) to OracleAS 10g (10.1.2.0.2).

14.4 Patching an Existing OracleAS Disaster Recovery Environment

For information about how to patch your OracleAS Disaster Recovery environment (patching OracleAS Guard 10.1.2.n.n (where n.n represents 0.0 and 0.2) with Release 10.1.3.0.0) to take advantage of the features in this latest release of OracleAS Guard, see the platform specific *Oracle Application Server Installation Guide* and specifically the chapter entitled "Installing in High Availability Environments: OracleAS Disaster Recovery." This chapter contains a section entitled "Patching OracleAS Guard Release 10.1.2.n.n with Release 10.1.3.0.0" that describes this patching process.

15

Setting Up a DNS Server

This chapter provides instructions on setting up a DNS server in UNIX. These instructions are applicable for setting up the site-specific DNS zones used for hostname resolution in the example in Figure 11–7, "DNS Resolution Topology Overview".

> **Note:** The DNS setup information provided in this chapter is an example to aid in the understanding of OracleAS Disaster Recovery operations. It is generic to DNS, and other appropriate DNS documentation should be consulted for comprehensive DNS information.

For the discussion in this chapter, the DNS server that is set up creates and services a new DNS zone with the unique domain `oracleas`. Within the zone, this DNS server resolves all requests for the `oracleas` domain and forwards other requests to the overall wide area company DNS server(s).

On the UNIX host that will act as the DNS zone server, perform the following steps:

1. Create the name server configuration file `/var/named.conf`. Assuming the wide area company DNS server IP address is 123.1.15.245, the contents of this file should be as follows:

```
options {
          directory "/var/named";
          forwarders {
           123.1.15.245;
         };
};

zone "." in {
              type hint;
              file "named.ca";
};

zone "oracleas" {
                  type master;
                  file "oracleas.zone";
};

zone "0.0.127.IN-ADDR.ARPA {
                           type master;
                           file "127.zone";
};
```

2. Create the root hint file `/var/named/named.ca`, which has the following contents (123.1.2.117 is the IP of the zone DNS server):

```
.        999999   IN    NS    ourroot.private.
ourroot.private.   IN    A     123.1.2.117
```

3. Create the loopback address file `/var/named/127.zone`, which has the following contents (assume the zone DNS server's hostname is `aszone1`):

```
$ORIGIN    0.0.127.IN-ADDR.ARPA.
0.0.127.IN-ADDR.ARPA.    IN    SOA  aszone1.oracleas.  root.aszone1.oracleas.
(
           25           ; serial number
           900          ; refresh
           600          ; retry
           86400        ; expire
           3600       ) ; minimum TTL

0.0.127.IN-ADDR.ARPA.    IN    NS    aszone1.oracleas.
1                        IN    PTR   localhost.oracleas.
```

4. Create the zone data file `/var/named/oracleas.dns`, which has the following contents (values shown are applicable to the example of the production site in Figure 11–7):

```
;
;  Database file oracleas.dns for oracleas zone.
;    Zone version:  25
;
$ORIGIN oracleas.
oracleas.       IN    SOA    aszone1.oracleas.  root.aszone1.oracleas (
                25        ; serial number
                900       ; refresh
                600       ; retry
                86400     ; expire
                3600    ) ; minimum TTL

;
;    Zone NS records
;
oracleas.       IN        NS    aszone1.oracleas.

;
;    Zone records
;
localhost       IN     A     127.0.0.1

asmid1          IN     A     123.1.2.333
asmid2          IN     A     123.1.2.334
infra           IN     A     123.1.2.111
remoteinfra     IN     A     213.2.2.210
```

5. Run the following command to start the name server:

```
/sbin/in.named
```

6. On all the hosts in the domain that is serviced by this DNS server, edit the `domain` and `nameserver` settings in the file `/etc/resolv.conf` as follows (all previous `nameserver` settings should be removed; 123.1.2.117 is assumed to the zone DNS server's IP address):

```
domain    oracleas
nameserver 123.1.2.117
```

16

Secure Shell (SSH) Port Forwarding

This chapter describes how secure shell (SSH) port forwarding may be used with Oracle Data Guard.

16.1 SSH Port Forwarding

OracleAS Guard automates the use of Oracle Data Guard, which sends redo data across the network to the standby system using Oracle Net-. SSH tunneling may be used with Oracle Data Guard as an integrated way to encrypt and compress the redo data before it is transmitted by the production system and subsequently decrypt and uncompress the redo data when it is received by the standby system.

See Also:

- Implementing SSH port forwarding with Data Guard: http://metalink.oracle.com/metalink/plsql/showdoc?db=NOT&id=225633.1
- Troubleshooting Data Guard network issues: http://metalink.oracle.com/metalink/plsql/showdoc?db=NOT&id=241925.1

Part V

Appendices

The information in this part is supplementary to the previous chapters of the book and is organized into the following appendixes:

- Appendix A, "Troubleshooting High Availability"
- Appendix B, "OracleAS Guard Error Messages"

A

Troubleshooting High Availability

This appendix describes common problems that you might encounter when deploying and managing Oracle Application Server in high availability configurations, and explains how to solve them. It contains the following topics:

- Section A.1, "Troubleshooting Active-Active Topologies"
- Section A.2, "Troubleshooting Active-Passive Topologies"
- Section A.3, "Troubleshooting OracleAS Disaster Recovery Topologies"
- Section A.4, "Need More Help?"

A.1 Troubleshooting Active-Active Topologies

Topics:

- Section A.1.1, "Registering an Application using ssoreg Fails"
- Section A.1.2, "OC4J_SECURITY Instance Fails to Start"
- Section A.1.3, "Logging into OracleAS Single Sign-On Takes a Long Time"
- Section A.1.4, "Oracle Internet Directory Does Not Start Up on One of the Nodes"
- Section A.1.5, "Unable to Connect to Oracle Internet Directory, and Oracle Internet Directory Cannot Be Restarted"
- Section A.1.6, "Cluster Configuration Assistant Fails During Installation"
- Section A.1.7, "odisrv Process Does Not Fail Over After "opmnctl stopall""
- Section A.1.8, "Oracle Internet Directory Processes Shut Down by OID Monitor"
- Section A.1.9, "Oracle Internet Directory Connections Being Disconnected by the Load Balancer or Firewall"

A.1.1 Registering an Application using ssoreg Fails

Problem

In high availability topologies where OracleAS Single Sign-On and Oracle Delegated Administration Services components are clustered in OracleAS Clusters, you get the following error message when you try to register an application using `ssoreg.sh` (`ssoreg.bat` on Windows):

```
java.io.EOFException
null
java.io.EOFException
        at
```

```
java.io.ObjectInputStream$BlockDataInputStream.peekByte(ObjectInputStream.java:243
5)
        at java.io.ObjectInputStream.readObject0(ObjectInputStream.java:1245)
        at java.io.ObjectInputStream.readObject(ObjectInputStream.java:324)
        at oracle.ias.sysmgmt.task.TaskMaster.daemon_exec(Unknown Source)
        at oracle.ias.sysmgmt.task.TaskMaster.remote_operation(Unknown Source)
        at oracle.ias.sysmgmt.cmdline.DcmCmdLine.ssoPropagate(Unknown Source)
        at oracle.ias.sysmgmt.cmdline.DcmCmdLine.execute(Unknown Source)
        at oracle.ias.sysmgmt.cmdline.DcmCmdLine.main(Unknown Source)
--end of dcmctl's output to stderr
Thu May 18 20:02:24 PDT 2006  dcmctl returned exit value 1
Thu May 18 20:02:24 PDT 2006  dcmctl returned unsuccessfully, exitValue 1
Thu May 18 20:02:24 PDT 2006  SSO registration tool failed.  Please check the
error in this log file, correct the problem and re-run the tool.
```

Solution

When you run `ssoreg.sh` (`ssoreg.bat` on Windows) in an environment where OracleAS Single Sign-On instances are in an OracleAS Cluster, you must ensure that the DCM daemon is running on all nodes in the cluster. This is because `ssoreg` invokes the DCM daemon on the node where you are running the command, and the DCM daemon needs to communicate with the other DCM daemons on the other nodes in the cluster.

To check that the DCM daemon is running, run the following command on all nodes in the cluster:

```
ORACLE_HOME/opmn/bin/opmnctl status
```

To start the DCM daemon, run the following command on nodes where the DCM daemon is not already running:

```
ORACLE_HOME/opmn/bin/opmnctl startproc ias-component=dcm-daemon
```

A.1.2 OC4J_SECURITY Instance Fails to Start

Problem

You are running Oracle Internet Directory in an Oracle RAC environment, and the OC4J_SECURITY instance fails to start. You see the following message in the `oidctl.log` file:

```
[gsdsiConnect] ORA-1017, ORA-01017: invalid username/password; logon denied.
```

Solution

Check that the `oidpwdlldap1` file is the same on all nodes in the Oracle RAC environment. You may have forgotten to copy the `oidpwdlldap1` file to all Oracle RAC nodes after running `oidpasswd` to change the ODS password. See Section 9.6, "About Changing the ODS Password on an Oracle RAC System" for details.

A.1.3 Logging into OracleAS Single Sign-On Takes a Long Time

Problem

Logging into OracleAS Single Sign-On might take a long time if you are running OracleAS Single Sign-On and Oracle Internet Directory on opposite sides of a firewall

(OracleAS Single Sign-On is running outside the firewall and Oracle Internet Directory inside the firewall) and if the firewall is configured to drop idle connections or recycle connections after the configured timeout period has elapsed.

Solution

1. Set the timeout on OracleAS Single Sign-On connections to a value smaller than the firewall and load balancer timeout values. The OracleAS Single Sign-On server will remove connections that are idle for longer than the specified value.

   You specify this value (in minutes) using the `connectionIdleTimeout` parameter in the *`ORACLE_HOME`*`/sso/conf/policy.properties` file. For example, the following line sets the timeout value for 20 minutes. The OracleAS Single Sign-On server will remove connections that are idle for longer than 20 minutes.

   ```
   connectionIdleTimeout = 20
   ```

   Restart the OC4J server (OC4J_SECURITY) that is running the OracleAS Single Sign-On server for the new value to take effect.

2. Set the timeout for database connections in the `SQLNET.EXPIRE_TIME` parameter in the *`ORACLE_HOME`*`/network/admin/sqlnet.ora` file. You also set this value to a value smaller than the firewall and load balancer timeout values.

   This parameter specifies how often the database server sends a probe packet to the client (which is the OracleAS Single Sign-On server). This periodic activity by the probe packet enables the OracleAS Single Sign-On server-to-database connections to stay active.

   The value is specified in minutes. In the following example, the database server sends the probe packet every 20 minutes to the client.

   ```
   SQLNET.EXPIRE_TIME = 20
   ```

   Restart the database for the new value to take effect.

Explanation: The firewall or load balancer might drop connections to Oracle Internet Directory and the database if the connections are idle for a certain time. When the firewall or load balancer drops a connection, it might not send a tcp close notification to the OracleAS Single Sign-On server. The OracleAS Single Sign-On server then is unaware that the connection is no longer valid and tries to use it to perform Oracle Internet Directory or database operations. When the OracleAS Single Sign-On server does not get a response, it tries the next connection. Eventually it tries all the connections in the pool before making fresh connections to Oracle Internet Directory or to the database.

By setting the timeout on the OracleAS Single Sign-On server and on the database to a value smaller than the timeout on the firewall or load balancer, you ensure that the connections are valid.

A.1.4 Oracle Internet Directory Does Not Start Up on One of the Nodes

Problem

If the time difference between the nodes in the OracleAS Cluster (Identity Management) is greater than 250 seconds, the OID Monitor will stop Oracle Internet Directory on the node that is behind. For example, if the time on node A is ahead of node B's by more than 250 seconds, then the OID Monitor will stop Oracle Internet Directory processes on node B.

For details on how OID Monitor works, see Section 3.7.2, "OID Monitor Details".

For details on time synchronization, see Section 3.7.2.2, "Time Discrepancy Between Nodes".

Solution

Synchronize the time on all nodes to within 250 seconds of each other.

A.1.5 Unable to Connect to Oracle Internet Directory, and Oracle Internet Directory Cannot Be Restarted

Problem

This issue applies only to Windows 2000 platforms. This issue has two symptoms:

Symptom #1: If you have configured your load balancer to monitor the Oracle Internet Directory ports using TCP port monitoring, you might see the "maximum number of connections reached" error in the Oracle Internet Directory log file. This means that clients are unable to connect to Oracle Internet Directory.

Symptom #2: If Oracle Internet Directory terminates, you are not able to restart it. When you try to restart it, you get a message that Oracle Internet Directory is unable to access its ports because the System Idle Process is already using them. Oracle Internet Directory needs exclusive access to its ports.

Solution

This problem is caused by an application (in this case, the load balancer) that performs TCP port monitoring on the Oracle Internet Directory ports. In TCP port monitoring, the application opens and closes connections to the Oracle Internet Directory ports. In Windows 2000, the connection is not closed properly; this is why you reach the maximum number of connections.

The workaround is not to use TCP port monitoring for the Oracle Internet Directory ports. Instead, use LDAP or HTTP port monitoring.

A.1.6 Cluster Configuration Assistant Fails During Installation

Problem

During the installation of distributed Oracle Identity Management topologies, the OracleAS Single Sign-On and Oracle Delegated Administration Services components are installed on their own nodes separate from the other Oracle Identity Management components. The Cluster Configuration Assistant may attempt to cluster the two resulting OracleAS Single Sign-On/Oracle Delegated Administration Services instances together. However, the error message "Instances containing disabled components cannot be added to a cluster" may appear. This message appears because Enterprise Manager cannot cluster instances with disabled components.

Solution

If the Cluster Configuration Assistant fails, you can cluster the instance after installation. In this case, to cluster the instance, you must use the "`dcmctl joincluster`" command instead of Application Server Control Console. You cannot use Application Server Control Console in this case because it cannot cluster instances that contain disabled components. In this case, the "home" OC4J instance is disabled.

A.1.7 odisrv Process Does Not Fail Over After "opmnctl stopall"

Problem

In OracleAS Cluster (Identity Management) and distributed OracleAS Cluster (Identity Management) topologies, when `opmnctl stopall` is executed to stop all OPMN-managed processes on that node, `odisrv` is not started automatically on the second node because `opmnctl stopall` is a normal administrative shutdown, not an actual node failure. In a true node failure, `odisrv` is started on the remaining node upon death detection of the original `odisrv` process.

Solution

If planned maintenance is required for OracleAS Cluster (Identity Management) and distributed OracleAS Cluster (Identity Management) topologies, use the `oidctl` command to explicitly stop and start `odisrv`.

On the node where `odisrv` is running, use the following command to stop it:

```
ORACLE_HOME/bin/oidctl connect=<dbConnect> server=odisrv inst=1 stop
```

On the remaining active node, start `odisrv` using the following command:

```
ORACLE_HOME/bin/oidctl connect=<dbConnect> server=odisrv instance=1
     flags="host=OIDhost port=OIDport" start
```

See Section 3.7.2.1, "Normal Shutdown vs. Process Failure" for details.

A.1.8 Oracle Internet Directory Processes Shut Down by OID Monitor

Problem

Oracle Internet Directory processes on one node are shut down by OID Monitor.

Solution

In active-active topologies, OID Monitor checks the time on each node running Oracle Internet Directory processes. If it discovers that the time difference between the nodes is more than 250 seconds, it shuts down the processes on the node that is behind in time.

To fix this, reset the time on the nodes such that the time on all nodes is within 250 seconds of each other. OID Monitor will detect the updated times and start up the Oracle Internet Directory processes.

See Section 3.7.2.2, "Time Discrepancy Between Nodes" for details.

A.1.9 Oracle Internet Directory Connections Being Disconnected by the Load Balancer or Firewall

Problem

The load balancer or firewall terminates connections to Oracle Internet Directory, and further connections from OC4J to Oracle Internet Directory cannot be made.

Solution

To fix this, set the `orclLDAPConnTimeout` attribute (in the "`cn=dsaconfig, cn=configsets, cn=oracle internet directory`" entry) to a value smaller than the "idle connection timeout" value configured on the load balancer or firewall.

This prevents the load balancer or firewall from terminating connections to Oracle Internet Directory.

The `orclLDAPConnTimeout` attribute is expressed in minutes.

Note that in this release and also in the 10.1.2.2.0 patch set, the `orclLDAPConnTimeout` attribute is independent of the `orclStatsPeriodicity` attribute when Oracle Internet Directory calculates the idle time of a connection.

However, in previous releases (releases 9.0.4.2, 9.0.4.3, 10.1.2.0, 10.1.2.0.2, and 10.1.2.1), Oracle Internet Directory takes into account the values for both attributes when it calculates the idle time. For these releases, you need to set the attributes as follows:

- Set the `orclStatsPeriodicity` attribute to a value less than half of the "idle connection timeout" value configured on the load balancer or firewall.
- Set the `orclLDAPConnTimeout` attribute to a value less than the "idle connection timeout" value configured on the load balancer or firewall.

The attribute values are expressed in minutes.

The values of the `orclStatsFlag` and `orclMaxTcpIdleConnTime` attributes are not used here.

For example, assume that the "idle connection timeout" value on the load balancer or firewall is set at 15. In this case, you can set the `orclStatsPeriodicity` attribute to 7 (which is less than half of 15) and the `orclLDAPConnTimeout` attribute to 12 (which is less than 15).

The `orclLDAPConnTimeout` attribute is in the "`cn=dsaconfig, cn=configsets, cn=oracle internet directory`" entry, while the other attributes are in the root DSE entry.

A.2 Troubleshooting Active-Passive Topologies

Topics:

A.2.1 Unable to Perform Online Database Backup and Restore in OracleAS Cold Failover Cluster Environment

Problem

Unable to perform online recovery of Infrastructure database due to dependencies and cluster administrator trying to bring the database down and then up during the recovery phase by the Backup and Recovery Tool.

Solution 1

To perform a clean recovery, use the following steps:

1. Bring all resources offline using the cluster administrator (for Windows, use Oracle Fail Safe).
2. Perform a normal shutdown of the Infrastructure database.
3. Start only the database service. You can do this from the Windows Service Manager, or you can run the following command:

```
net start OracleService<SID>
```

4. Run the Backup and Recovery Tool to perform the recovery of the database.

Solution 2

For Windows, the following steps can be used to perform a recovery:

1. In Oracle Fail Safe, under "Cluster Resources", select "ASDB (DB Resource)" in the Database tab.
2. For "Database Polling", select "Disabled" from the drop down list.
3. Using the Backup and Recovery Tool, perform an online restore of the Infrastructure database.

The database is not accessible for a brief period while the Backup and Recovery Tool stops and starts the database. Once the database starts up, it can be accessed by middle-tier and Infrastructure components.

A.2.2 Cannot Connect to Database for Restoration (Windows)

Problem

When you stop the OracleAS Metadata Repository database using Microsoft Cluster Administrator, Microsoft Cluster Administrator performs the strictest and fastest abort to shut down the database service. After the shutdown, you are unable to connect to the database.

The following steps illustrate the problem:

1. Access an OracleAS Metadata Repository that is used for testing.
2. Corrupt a database file (note: do not modify the `ts$` table).
3. Issue a SQL query to ensure that the database is corrupted.
4. Using Microsoft Cluster Administrator, verify that the database is online.
5. Using Oracle Fail Safe Manager, disable database polling.
6. Using Microsoft Cluster Administrator, take the database offline. This also takes OPMN and Application Server Control Console offline as they are dependencies of the database.
7. Try connecting as `sysdba`. The connection should fail.

At this time, you are unable to connect to the database to run backup/restore scripts to restore the database to a good version (because you corrupted a database file in step 2 above).

Solution

Use Oracle Fail Safe Manager (instead of Microsoft Cluster Administrator) to shut down the database. To do so:

1. In the Oracle Fail Safe Manager, right-click the "ASDB" resource (default if not changed), and select "Immediate".
2. Start the database service using Windows Service Manager.
3. Connect to the database as `sysdba`. The connection should be successful.

A.3 Troubleshooting OracleAS Disaster Recovery Topologies

This section describes common problems and solutions in OracleAS Disaster Recovery configurations. It contains the following topics:

A.3.1 Standby Site Not Synchronized

In the OracleAS Disaster Recovery standby site, you may find that the site's OracleAS Metadata Repository is not synchronized with the OracleAS Metadata Repository in the primary site.

Problem

The OracleAS Disaster Recovery solution requires manual configuration and shipping of data files from the primary site to the standby site. Also, the data files (archived database log files) are not applied automatically in the standby site, that is, OracleAS Disaster Recovery does not use managed recovery in Oracle Data Guard.

Solution

The archive log files have to be applied manually. The steps to perform this task is found in Chapter 11, "OracleAS Disaster Recovery".

A.3.2 Failure to Bring Up Standby Instances After Failover or Switchover

Standby instances are not started after a failover or switchover operation.

Problem

IP addresses are used in instance configuration. OracleAS Disaster Recovery setup does not require identical IP addresses in peer instances between the production and standby site. OracleAS Disaster Recovery synchronization does not reconcile IP address differences between the production and standby sites. Thus, if you use explicit IP address xxx.xx.xxx.xx in your configuration, the standby configuration after synchronization will not work.

Solution

Avoid using explicit IP addresses. For example, in OracleAS Web Cache and Oracle HTTP Server configurations, use ANY or host names instead of IP addresses as listening addresses

A.3.3 Switchover Operation Fails At the Step dcmctl resyncInstance -force -script

The OracleAS Disaster Recovery asgctl switchover operation requires that the value of the TMP variable be defined the same in the `opmn.xml` file on both the primary and standby sites.

Problem

OracleAS Disaster Recovery switchover fails at the step dcmctl resyncInstance -force -script and displays a message that a directory could not be found.

Solution

During a switchover operation, the opmn.xml file is copied from the primary site to the standby site. For this reason, the value of the TMP variable must be defined the same in the `opmn.xml` file on both primary and standby sites; otherwise, the switchover operation will fail. Make sure the TMP variable is defined identically in the opmn.xml files and resolves to the same directory structure on both sites before attempting to perform an asgctl switchover operation.

For example, the following code snippets for a Windows and UNIX environment show a sample definition of the TMP variable.

```
Example in Windows Environment:
-------------------------------
.
.
.
<ias-instance id="infraprod.iasha28.us.oracle.com">
 <environment>
 <variable id="TMP" value="C:\DOCUME~1\ntregres\LOCALS~1\Temp"/>
 </environment>
.
.
.
Example in Unix Environment:
----------------------------
.
.
.
<ias-instance id="infraprod.iasha28.us.oracle.com">
 <environment>
 <variable id="TMP" value="/tmp"/>
 </environment>
.
.
.
```

A workaround to this problem is to change the value of the TMP variable in the `opmn.xml` file on the primary site, perform a dcmctl update config operation, then perform the asgctl switchover operation. This approach saves you having to reinstall the mid-tiers to make use of an altered TMP variable.

A.3.4 Unable to Start Standalone OracleAS Web Cache Installations at the Standby Site

OracleAS Web Cache cannot be started at the standby site possibly due to misconfigured standalone OracleAS Web Cache after failover or switchover.

Problem

OracleAS Disaster Recovery synchronization does not synchronize standalone OracleAS Web Cache installations.

Solution

Use the standard Oracle Application Server full CD image to install the OracleAS Web Cache component

A.3.5 Standby Site Middle-tier Installation Uses Wrong Hostname

A middle-tier installation in the standby site uses the wrong hostname even after the machine's physical hostname is changed.

Problem

Besides modifying the physical hostname, you also need to put it as the first entry in `/etc/hosts` file. Failure to do the latter will cause the installer to use the wrong hostname.

Solution

Put the physical hostname as the first entry in the `/etc/hosts` file. See Section 11.2.2, "Configuring Hostname Resolution" on page 11-15 for more information.

A.3.6 Failure of Farm Verification Operation with Standby Farm

When performing a verify farm with standby farm operation, the operation fails with an error message indicating that the middle-tier machine instance cannot be found and that the standby farm is not symmetrical with the production farm.

Problem

The verify farm with standby farm operation is trying to verify that the production and standby farms are symmetrical to one another, that they are consistent, and conform to the requirements for disaster recovery.

The verify operation is failing because it sees the middle-tier instance as `mid_tier.<hostname>` and not as `mid_tier.<physical_hostname>`. You might suspect that this is a problem with the environmental variable `_CLUSTER_NETWORK_NAME_`, which is set during installation. However, in this case, it is not because a check of the `_CLUSTER_NETWORK_NAME_` environmental variable setting finds this entry to be correct. However, a check of the contents of the `/etc/hosts` file, indicates that the entries for the middle tier in question are incorrect. That is, all middle-tier installations take the hostname from the second column of the `/etc/hosts` file.

For example, assume the following scenario:

- Two environments are used: `examp1` and `examp2`
- OracleAS Infrastructure (Oracle Identity Management and OracleAS Metadata Repository) is first installed on `examp1` and `examp2` as host `infra`
- OracleAS middle-tier (OracleAS Portal and OracleAS Wireless) is then installed on `examp1` and `examp2` as host `node1`
- Basically, these are two installations (OracleAS Infrastructure and OracleAS middle-tier) on a single node
- Updated the latest `duf.jar` and `backup_restore` files on all four Oracle homes

- Started OracleAS Guard (`asgctl`) on all four Oracle homes (OracleAS Infrastructure and OracleAS middle-tier on two nodes)
- Performed `asgctl` operations: `connect asg`, `set primary`, `dump farm`
- Performed `asgctl verify farm` with `standby farm` operation, but it fails because it sees the instance as `mid-tier.examp1` and not as `mid_tier.node1.us.oracle.com`

A check of the `/etc/hosts` file shows the following entry:

```
123.45.67.890 examp1 node1.us.oracle.com node1 infra
```

Then `ias.properties` and farms shows the following and the verify operation is failing:

```
IASname=midtier_inst.examp1
```

However, the `/etc/hosts` file should actually be the following:

```
123.45.67.890 node1.us.oracle.com node1 infra
```

Then `ias.properties` and farms shows the following and the verify operation succeeds:

```
IASname=midtier_inst.node1.us.oracle.com
```

Solution

Check and change the second column entry in your `/etc/hosts` file to match the hostname of the middle-tier node in question as described in the previous explanation.

A.3.7 Sync Farm Operation Returns Error Message

A `sync farm to` operation returns the error message: "Cannot Connect to asdb"

Problem

Occasionally, an administrator may forget to set the primary database using the `asgctl` command line utility in performing an operation that requires that the asdb database connection be established prior to an operation. The following example shows this scenario for a `sync farm to` operation:

```
ASGCTL> connect asg hsunnab13 ias_admin/iastest2
Successfully connected to hsunnab13:7890
ASGCTL>
.
.
.
<Other asgctl operations may follow, such as verify farm, dump farm,
<and show operation history, and so forth that do not require the connection
<to the asdb database to be established or a time span may elapse of no activity
<and the administrator may miss performing this vital command.
.
.
.
ASGCTL> sync farm to usunnaa11
prodinfra(asr1012): Syncronizing each instance in the farm to standby farm
prodinfra: -->ASG_ORACLE-300: ORA-01031: insufficient privileges
prodinfra: -->ASG_DUF-3700: Failed in SQL*Plus executing SQL statement:  connect
null/******@asdb.us.oracle.com as sysdba;.
prodinfra: -->ASG_DUF-3502: Failed to connect to database asdb.us.oracle.com.
```

```
prodinfra: -->ASG_DUF-3504: Failed to start database asdb.us.oracle.com.
prodinfra: -->ASG_DUF-3027: Error while executing Syncronizing each instance in
the farm to standby farm at step - init step.
```

Solution

Perform the `asgctl set primary database` command. This command sets the connection parameters required to open the asdb database in order to perform the `sync farm to` operation. Note that the `set primary database` command must also precede the `instantiate farm to` command and `switchover farm to` command if the primary database has not been specified in the current connection session.

A.3.8 On Windows Systems Use of asgctl startup Command May Fail If the PATH Environment Variable Has Exceeded 1024 Characters

On Windows systems, if your system PATH environment variable has exceeded the 1024 character limit because you have many OracleAS instances installed or many third party software installations, or both on your system, the asgctl startup command may fail because you are starting the OracleAS Guard server outside of OPMN and the system cannot resolve the directory path.

Problem

Occasionally, on Windows systems with many installations, OracleAS instances or third party software, or both, the asgctl startup command, which is run outside of OPMN, may return a popup error stating it could not find a dynamic link library for a particular file, `orawsec9.dll`, followed by a DufException. For example:

```
C:\product\10.1.3\OC4J_1\dsa\bin> asgctl startup
<<Popup Error:>>
The dynamic link library *orawsec9.dll* could not be found.
<<The exception:>>
oracle.duf.DufException
        at oracle.duf.DufOsBase.constructInstance(DufOsBase.java:1331)
        at oracle.duf.DufOsBase.getDufOs(DufOsBase.java:122)
        at
oracle.duf.DufHomeMgr.getCurrentHomePath(DufHomeMgr.java:582)
        at oracle.duf.dufclient.DufClient.main(DufClient.java:132)
stado42: -->ASG_SYSTEM-100: oracle.duf.DufException
--------------------------------------------------------------------------------
```

However, this dll does exist in the ORACLE_HOME\bin directory.

This error is not seen in OracleAS Guard standalone kit because the file `orawsec9.dll` exists in the ORACLE_HOME\dsa\bin folder.

Solution

The workaround is to either manually edit the system PATH variable with the required path information or manually override the PATH in the command prompt by specifying the relevant %PATH% variables. For example:

```
C:\set PATH=C:\product\10.1.3\OracleAS_OC4J_2\bin;
C:\product\10.1.3\OracleAS_OHS1\jre\1.4.2\bin\client;
C:\product\10.1.3\OracleAS_OHS1\jre\1.4.2\bin;
C:\product\10.1.3\OracleAS_OHS1\bin;C:\product\10.1.3\OC4J_1\bin

C:\product\10.1.3\OC4J_1\dsa\bin> asgctl startup
```

A.4 Need More Help?

In case the information in the previous section is not sufficient, you can find more solutions on Oracle *MetaLink*, http://metalink.oracle.com. If you do not find a solution for your problem, log a service request.

See Also:

- *Oracle Application Server Release Notes*, available on the Oracle Technology Network: http://www.oracle.com/technology/documentation/index.html

B

OracleAS Guard Error Messages

The following sections describe the OracleAS Guard error messages. Though not shown, OracleAS Guard error messages are preceded by an ASG prefix. Error messages are categorized into the following groups and subgroups:

B.1 DGA Error Messages

The following are DGA error messages.

> **Note:** The symbols {0}, {1}, and {2} are variables that will be replaced by the name of the object.

12001, Error while creating a DGA template.

Cause: An error occurred while creating a template file.

Action: See secondary error.

12500, Standby database instance {0} already exists on host {1}.

Cause: The standby database instance specified already exists on target host.

Action: Either select a new instance or remove the current instance.

B.1.1 LRO Error Messages

The following are LRO error messages.

13000, Error during Create Physical Standby: Prepare-init.

Cause: Error occurred during specified step.

Action: See secondary error.

13001, Error during Create Physical Standby: Prepare-check standby.

Cause: Error occurred during specified step.

Action: See secondary error.

13002, Error during Create Physical Standby: Prepare-primary processing.

Cause: Error occurred during specified step.

Action: See secondary error.

13003, Error during Create Physical Standby: Prepare-standby processing.

Cause: Error occurred during specified step.

Action: See secondary error.

13004, Error during Create Physical Standby: Prepare-sqlnet configuration.

Cause: Error occurred during specified step.

Action: See secondary error.

13005, Error during Create Physical Standby: Copy-init.

Cause: Error occurred during specified step.

Action: See secondary error.

13006, Error during Create Physical Standby: Copy-validate standby.

Cause: Error occurred during specified step.

Action: See secondary error.

13007, Error during Create Physical Standby: Copy-file copy.

Cause: Error occurred during specified step.

Action: See secondary error.

13008, Error during Create Physical Standby: Finish-init.

Cause: Error occurred during specified step.

Action: See secondary error.

13009, Error during Create Physical Standby: Finish-prepare primary.

Cause: Error occurred during specified step.

Action: See secondary error.

13010, Error during Create Physical Standby: Finish-configure primary.
Cause: Error occurred during specified step.
Action: See secondary error.

13011, Error during Create Physical Standby: Finish-configure standby.
Cause: Error occurred during specified step.
Action: See secondary error.

B.1.2 Undo Error Messages

The following are undo error messages.

13015, Error trying to Undo Create Physical Standby: Prepare.
Cause: Error occurred during undo of the prepare task.
Action: See secondary error.

13016, Error trying to Undo Create Physical Standby: Copy.
Cause: Error occurred during undo of the copy task.
Action: See secondary error.

13017, Error trying to Undo Create Physical Standby: Finish.
Cause: Error occurred during undo of the finish task.
Action: See secondary error.

B.1.3 Create Template Error Messages

The following are create template error messages.

13020, Error during Create Template: init.
Cause: Error occurred during specified step.
Action: See secondary error.

13021, Error during Create Template: primary processing.
Cause: Error occurred during specified step.
Action: See secondary error.

13022, Error during Create Template: standby processing.
Cause: Error occurred during specified step.
Action: See secondary error.

13023, Error during Create Template: finish.
Cause: Error occurred during specified step.
Action: See secondary error.

B.1.4 Switchover Physical Standby Error Messages

The following are switchover physical standby error messages.

13051, Error performing a physical standby switchover.
Cause: Error occurred in performing a switchover.
Action: See secondary error.

13052, The primary database is not in the proper state to perform a switchover.

Cause: The switchover status of the primary database must be either "TO STANDBY" or "SESSIONS ACTIVE".

Action: Make sure the SWITCHOVER_STATUS of the V$DATABASE table is either "TO STANDBY" or "SESSIONS ACTIVE".

13053, The standby database is not in the proper state to perform a switchover.

Cause: The switchover status of the standby database must be either "TO PRIMARY" or "SWITCHOVER PENDING".

Action: Make sure the SWITCHOVER_STATUS of the V$DATABASE table is either "TO PRIMARY" or "SWITCHOVER PENDING".

13504, Error switching the database role from primary to standby.

Cause: Failed to switchover database role from primary to standby.

Action: See secondary error.

13505, Error switching the database role from standby to primary.

Cause: Failed to switchover database role from standby to primary.

Action: See secondary error.

13061, Error failing over physical standby database.

Cause: Error occurred in performing a failover of a standby database.

Action: See secondary error.

B.2 Duf Error Messages

The following are Duf error messages.

3000, Server error {0}.

Cause: Invalid argument was supplied.

Action: Pass in a valid argument.

3001, Invalid argument {0}.

Cause: Invalid argument was supplied.

Action: Pass in a valid argument.

3002, Invalid log path {0}.

Cause: Invalid log path specification.

Action: Specify a valid log path.

3003, Invalid command line value {0} specified.

Cause: Invalid command line specification.

Action: Correct the command line option and retry.

3004, Invalid command action {0} specified.

Cause: Invalid command action specification.

Action: Correct the command line action and retry.

3005, Invalid command argument {0} specified, commands must begin with a hyphen.

Cause: Command argument did not start with a hyphen.

Action: Enter a correct command line argument.

3006, Command line argument {0} missing a required value.

Cause: Command argument missing a required value.

Action: Enter a correct command line argument value.

3007, Command line argument {0} given an incorrect value {1}.

Cause: Command argument value is incorrect.

Action: Enter a correct command line argument value.

3008, Command line argument {0} is required but missing.

Cause: Command argument value is missing.

Action: Enter a correct command line argument value.

3009, Invalid session ID.

Cause: The client passed an invalid session ID.

Action: Enter a correct command line argument value.

3010, Duplicate session ID.

Cause: The session ID is already in use.

Action: Enter a correct command line argument value.

3011, Unsatisfied link error for {0} in library DufNatives.

Cause: An attempt to make a call using the DufNatives library failed.

Action: Make sure the DufNatives library is correctly installed.

3012, Checksum error in password.

Cause: The login password has a checksum error.

Action: Try to reconnect.

3013, Operation failed.

Cause: The specified operation failed.

Action: See secondary error.

3014, Invalid command line specified.

Cause: Invalid command line specification.

Action: Correct command line option and retry.

3015, Error getting local host name.

Cause: Error trying to get local host name.

Action: See secondary error.

3016, No encrypt key.

Cause: No encrypt key supplied. Encryption requires an encrypt key.

Action: This is an internal programming error.

3017, Error encrypting data.

Cause: Failed to encrypt the given data.

Action: See secondary error.

3018, Error missing plan for specified request {0}, cannot process.

Cause: Could not find plan for specified request.

Action: Either specify a valid request or supply valid plan.

3019, Server does not recognize application ID.

Cause: Client specified an application ID that the server does not support.

Action: Contact Oracle support.

3020, Failed to authenticate user {0}. Please enter the correct user name and password.

Cause: Client supplied incorrect OS user name or password or both.

Action: Make sure the correct OS user name and password are used.

3021, No user name and/or password are supplied for authentication.

Cause: Client did not supply a user name or password or both through the "connect duf" command.

Action: Make sure user issue a "connect duf" command before any other commands.

3022, Failed to authorize user {0}. User must have administrator privilege on the server system.

Cause: The user name provided by the client to connect to DUF server must have administrator privilege on the server system. This error is applicable on Windows system only.

Action: Make sure the user account belongs to the administrator group on the server system.

3023, Error: There is no connection to a DUF server.

Cause: You must connect to DUF server before issues other commands.

Action: Connect to the DUF server.

3024, Failed to authorize user {0}. The owner account of the Oracle Home must be used.

Cause: The user name provided by the client to connect to DUF server must be the same user from which the Oracle home is installed. This error is applicable on UNIX system only.

Action: Make sure the user account is the same as that of the Oracle home.

3025, The operation has been cancelled.

Cause: The operation has been cancelled by either the user or the DUF internal software.

Action: None.

3026, The {0} task must complete successfully before running the {1} task.

Cause: An attempt was made to run the specified task before the required previous task has successfully completed.

Action: Rerun the required previous task.

3027, Error while executing {0} at step - {1}.

Cause: Error during specified step of specified operation.

Action: Check secondary error.

3028, Failed to start DUF server on host {0}.

Cause: Error during specified step of specified operation.

Action: Check DUF log file for more information.

3029, Failed to start {0} server with exception.

Cause: Error trying to start the server.

Action: See secondary error.

3030, Error, cannot resolve host name {0}.

Cause: Error trying to resolve specified host name.

Action: Check that the host name is correctly specified.

3031, Error, Invalid user name {0}. Only {1} account can connect to a DSA server.

Cause: Only ias_admin can connect to a DSA server.

Action: Please use ias_admin to connect to a DSA server.

3032, Failed to start {0} server on host {1}. Start server on specified host and reconnect.

Cause: Error trying to start the server on the specified host while trying to connect.

Action: Start the server manually and retry the connect.

3033, Error, the server is shutting down.

Cause: Error communicating with the server.

Action: Retry the operation.

3034, Invalid command line specified: - {0}.

Cause: Invalid command line specification.

Action: Correct the command line option and retry.

3035, Failed to kill the OracleAS Guard (DSA) server process {0} with error {1}.

Cause: OracleAS Guard client is unable to kill the OracleAS Guard (DSA) server process.

Action: Use "kill -9 <pid>" command to kill the process from the command line prompt.

3100, Error reading file {0}.

Cause: Error trying to read from file.

Action: See secondary error.

3101, Error writing file {0}.

Cause: Error trying to write file.

Action: See secondary error.

3102, Error creating file {0}.

Cause: Error trying to create specified file.

Action: See secondary error.

3103, Error deleting file {0}.

Cause: Error trying to delete a file.

Action: See secondary error.

3104, Error opening file {0}.
Cause: Error trying to open file.
Action: See secondary error.

3105, File {0} not found.
Cause: Error trying to open file.
Action: See secondary error.

3106, No read access to file {0}.
Cause: Error trying to open file.
Action: See secondary error.

3107, No write access to file {0}.
Cause: Error trying to open file.
Action: See secondary error.

3108, File specification {0} must be absolute.
Cause: Error trying to open file.
Action: See secondary error.

3109, Error closing file {0}.
Cause: Error trying to close the file.
Action: See secondary error.

3110, Error creating dir {0}.
Cause: Error trying to create specified directory.
Action: See secondary error.

3111, Error deleting dir {0}.
Cause: Error trying to delete specified directory.
Action: See secondary error.

3112, Error expanding file wildcard specification {0}.
Cause: Error trying to process file wildcard specification.
Action: See secondary error.

3120, Error opening configuration file {0}.
Cause: Error trying to open configuration file.
Action: Make sure configuration file exists or specified correctly.

3121, Error creating zip file {0}.
Cause: Error trying to create a zip file.
Action: See secondary error.

3122, There are no files to be zipped.
Cause: The directory to be zipped has no files in it.
Action: Make sure the directory to be zipped has files in it.

3123, Error adding files in directory {0} to zip file.
Cause: Error adding files in the given directory to zip file.

Action: See secondary error.

3124, No zip file is specified.

Cause: No zip file is specified.

Action: Internal error.

3125, Error extracting files from zip file {0}.

Cause: Error extracting files from a zip file.

Action: See secondary error.

3400, Error processing XML document.

Cause: Error processing XML document.

Action: See secondary error.

3401, Error processing XML node.

Cause: Error processing XML node.

Action: See secondary error.

3402, Error parsing XML request message.

Cause: There was an error parsing the XML request message.

Action: Contact Oracle support.

3403, Error parsing XML response message.

Cause: There was an error parsing the XML response message.

Action: Contact Oracle support.

3404, Error parsing XML body string.

Cause: There was an error parsing the XML body string.

Action: Contact Oracle support.

3405, Error writing the body to an XML DOM.

Cause: There was an error writing the XML body string.

Action: Contact Oracle support.

3406, Error reading the body from an XML DOM.

Cause: There was an error reading the XML body string.

Action: Contact Oracle support.

3407, Error writing a work item to an XML DOM.

Cause: There was an error writing the XML body string.

Action: Contact Oracle support.

3408, Error reading a work item from an XML DOM.

Cause: There was an error reading the XML body string.

Action: Contact Oracle support.

3409, Error parsing an XML string.

Cause: There was an error parsing the XML string.

Action: Contact Oracle support.

3410, Error converting XML DOM to string.

Cause: There was an error converting the DOM tree to a XML string.

Action: Contact Oracle support.

3411, Error reading XML DOM tree.

Cause: There was an error reading the XML DOM tree.

Action: Contact Oracle support.

B.2.1 Database Error Messages

The following are database error messages.

3501, Failed to initialize DufDb class.

Cause: There was an error creating the DufDb class.

Action: See secondary error.

3502, Failed to connect to database {0}.

Cause: There was an error connecting to the database.

Action: See secondary error.

3503, Failed to verify database {0}.

Cause: There was an error verifying the database.

Action: See secondary error.

3504, Failed to start database {0}.

Cause: There was an error starting the database.

Action: See secondary error.

3505, Failed to create pfile to include spfile.

Cause: There was an error creating the given pfile.

Action: See secondary error.

3506, Failed to turn on archivelog mode for the database.

Cause: There was an error turning on archivelog mode.

Action: See secondary error.

3507, Failed to create the standby database control file.

Cause: There was an error creating the standby database control file.

Action: See secondary error.

3508, Failed to create the pfile.

Cause: There was an error creating the database init parameter file.

Action: See secondary error.

3509, Failed to create the spfile.

Cause: There was an error creating the database spfile.

Action: See secondary error.

3510, Output reader thread for {0} terminated.

Cause: The output reader thread is terminated.

Action: Contact Oracle support.

3511, Error creating local worker on node {0}.

Cause: This is an internal error.

Action: Contact Oracle support.

3512, Error creating remote worker on node {0}.

Cause: There is a problem communicating with the remote server.

Action: Make sure that the remote server is accessible.

3513, Database is not started {0}.

Cause: The specified database has not been started.

Action: Start the specified database.

3514, Failed to stop database {0}.

Cause: There was an error stopping the database.

Action: See secondary error.

3515, Failed querying database to determine current archivelog mode.

Cause: There was an error querying the database to determine current archive mode.

Action: See secondary error.

3516, Failed to query redo log information for database.

Cause: There was an error querying the database redo log information.

Action: See secondary error.

3517, Failed to drop standby redo log for database.

Cause: There was an error dropping the standby redo log.

Action: See secondary error.

3518, Failed to start managed recovery for standby database.

Cause: There was an error starting managed recovery for the standby database.

Action: See secondary error.

3519, Failed to cancel managed recovery for standby database.

Cause: There was an error cancelling managed recovery for the standby database.

Action: See secondary error.

3520, Failed to determine the existence of database instance.

Cause: There was an error determining the existence of the given database instance.

Action: See secondary error.

3521, Invalid database instance {0} specified in the template file; DUF found instance {1}.

Cause: The standby database instance specified in the template file is different from the one DUF found on the system.

Action: Please either rerun the prepare and copy phases with the new standby instance or specify the correct standby database instance found on the system.

3522, The pfile {0} needed to generate an spfile is missing.

Cause: A pfile needed to create the spfile used by the standby database is missing.

Action: Please either rerun the prepare and copy phases to generate the pfile or manually create one with the correct values.

3523, The standby database cannot have the same service name as the primary database.

Cause: The standby service name is the same as the primary.

Action: Change the standby service name.

3524, Error: The primary database is not set.

Cause: The primary database is not defined.

Action: Set the primary database first.

3525, Error: The standby database is not set.

Cause: The standby database is not defined.

Action: Set the standby database first.

3526, Set the primary database before setting the standby database.

Cause: The standby service name is the same as the primary on the same host.

Action: Change the standby service name.

3527, The database tablespace map is NULL.

Cause: This is an internal error.

Action: Contact Oracle support.

3528, Error initializing init parameter file {0}.

Cause: An error occurred trying to initialize the parameter file.

Action: See secondary error.

3529, Error writing init parameter file {0}.

Cause: An error occurred trying to write the parameter file.

Action: See secondary error.

3530, Error in setting the protection mode for database {0}.

Cause: An error occurred trying to set the protection mode.

Action: See secondary error.

3531, Error opening database in read only mode for database {0}.

Cause: An error occurred trying to open the database in read only mode.

Action: See secondary error.

3532, Failed to get init parameter value from {0}.

Cause: Error trying to get the parameter value from the init parameter file.

Action: See secondary error.

3533, No user name and/or password is specified for database {0}.

Cause: Error trying to get the parameter value from the init parameter file.

Action: User must specify the user name and password to be used to connect to the database using "set primary database" or "set standby database" command.

3534, The standby database cannnot have the same host as the primary database.

Cause: The standby host is the same as the primary.

Action: Change the standby or primary database host name.

3535, Failed to create standby redo log.

Cause: An error occurred trying to create a standby redo log.

Action: See secondary error.

3536, Failed to get a list of standby database(s) from log archive destination.

Cause: An error occurred trying to get a list of standby databases from the log archive destination parameters.

Action: See secondary error.

3537, Failed to add standby database as a log archive destination.

Cause: An error occurred trying to add a standby database as a log archive destination.

Action: See secondary error.

3538, Failed to remove standby database as a log archive destination.

Cause: An error occurred trying to remove a standby database as a log archive destination.

Action: See secondary error.

3539, Error: The new primary database is not set.

Cause: The new primary database is not defined.

Action: Set the new primary database first.

3540, Error processing template file {0}.

Cause: Error trying to process template file.

Action: Correct protection and retry operation.

3541, Invalid database protection specified in template file {0}.

Cause: Error trying to process protection value in template file.

Action: Correct protection and retry operation.

3542, Failed to query database role.

Cause: Error trying to query the database role.

Action: See secondary error.

3543, Error processing command, must be connected to a OracleAS Guard server in the primary topology.

Cause: User is connected to server on a topology that is not the primary topology.

Action: Connect to primary topology node.

3544, Error processing command, must be connected to a OracleAS Guard server in the standby topology.

Cause: User is connected to server on a topology that is not the standby topology.

Action: Connect to primary topology node.

3545, Error trying to remove old passwd file %1 while creating new db.

Cause: Could not delete the old password file as part of a delete database operation. This is a problem when trying to create a new database.

Action: Delete the stale password file.

3546, Error, database SID was expected to have value but it is empty.

Cause: The database SID was suppose to have a value but it is empty.

Action: This is an internal error.

3547, Error storing DB Credentials in the clipboard of the server.

Cause: Failed to store DB credentials in the clipboard on the specified server.

Action: Internal error.

3548, Error storing DB info in the clipboard of the server.

Cause: Failed to store DB information in the clipboard on the specified server.

Action: Internal error.

3549, Error cleaning up the database on the standby host.

Cause: Failed to clean up the database on the standby host.

Action: See secondary error.

3550, Failed to find a valid Oracle Home.

Cause: A valid Oracle home was not found for this operation.

Action: Create a valid Oracle home.

3551, Oracle Data Guard Home must have the same owner as the database server home.

Cause: The Oracle Data Guard Home is owned by a different user than the database server home.

Action: Reinstall Oracle Data Guard user from the owner of Oracle database server.

3552, Specified Oracle Home {0} could not be found.

Cause: The specified Oracle home could not be found.

Action: Please specify a valid Oracle home.

3553, An error occurred getting the list of Oracle Homes on the system.

Cause: The list of Oracle homes could not be read.

Action: Make sure the Oracle inventory is valid.

3554, The Oracle home that contains SID {0} cannot be found.

Cause: The Oracle home that contains a specific SID cannot be found.

Action: Make sure the Oracle home inventory is valid.

3555, Error accessing the Oracle home inventory. Make sure the inventory file exists.

Cause: The Oracle home inventory cannot be accessed.

Action: Make sure the Oracle home inventory exists

3556, Error: Unable to find the Oracle home within path {0}.

Cause: The Oracle home within the given path cannot be found.

Action: Make sure the Oracle home inventory exists.

B.2.2 Connection and Network Error Messages

The following are connection and network error messages.

3600, Error connecting to server: Unknown node {0}.

Cause: The server host is unknown to the client.

Action: Contact Oracle support.

3601, Error connecting to server node {0}.

Cause: The client cannot connect to the server.

Action: Contact Oracle support.

3602, File Copy protocol error.

Cause: There was an internal protocol error while copying files.

Action: Contact Oracle support

3603, Error sending data across network.

Cause: There was a network error.

Action: Retry operation.

3604, Error receiving data across network.

Cause: There was a network error.

Action: Retry operation.

3605, The file copy operation has been terminated.

Cause: The copy aborted due to an error.

Action: Retry operation.

3606, Error connecting to file copy server {0} on port {0}.

Cause: The copy server is not running.

Action: Contact Oracle support.

3607, Error opening file copy server socket on {0} with port {0}.

Cause: The copy aborted due to an error.

Action: Retry operation.

3608, Error connecting to clipboard.

Cause: There is no connection to the clipboard server.

Action: Retry operation.

3609, Error while copying {0} to {1}.

Cause: Error occurred during a file copy.

Action: See secondary error.

3610, Error starting online backup.

Cause: Error occurred while putting tablespace in online backup mode.

Action: See secondary error.

3611, Error ending online backup.

Cause: Error occurred while restore tablespace from online backup mode.

Action: See secondary error.

3612, Error listening on server port {0}.

Cause: Error occurred while listening on port.

Action: Check if server is already running.

3613, Network Buffer Overflow Detected.

Cause: The network protocol detected a buffer overflow due to a bug or attack.

Action: Call Oracle Support.

B.2.3 SQL*Plus Error Messages

The following are SQL*Plus error messages.

3700, Failed in SQL*Plus executing SQL statement: {0}.

Cause: Failed to execute the specified SQL statement.

Action: See secondary error.

3701, Failed starting SQL*Plus : {0}.

Cause: Failed to execute the specified SQL statement.

Action: See secondary error.

B.2.4 JDBC Error Messages

The following are JDBC error messages.

3751, Failed to register Oracle JDBC driver: oracle.jdbc.OracleDriver.

Cause: Failed to register the Oracle JDBC driver.

Action: Make sure that Oracle JDBC driver is installed on the local system.

3752, There is no JDBC connection to the database.

Cause: There is no connection to the database server.

Action: Connect to a database server first, then try the operation again.

3753, Failed to connect to the database.

Cause: Unable to connect to the database server.

Action: See secondary error.

3754, Failed to disconnect from the database.

Cause: Unable to disconnect from the database server.

Action: See secondary error.

3755, Failed to execute the SQL statement.

Cause: Failed to execute the SQL statement.

Action: See secondary error.

3756, Failed to run the SQL query.

Cause: Failed to run the SQL query statement.

Action: See secondary error.

3757, Failed to close the Oracle result set or the Statement object.

Cause: Failed to close the Oracle result set or the Statement object.

Action: See secondary error.

3758, This method can not be used to verify the physical standby database.

Cause: This is a programming error.

Action: Contact Oracle support.

3759, Verify DB query returned no data.

Cause: Verify database query returned no data.

Action: See secondary error.

3760, Failed to query the archive log destination information.

Cause: Failed to query the archive log destination information.

Action: See secondary error.

3761, Failed to query the redo log information.

Cause: Failed to query the redo log information.

Action: See secondary error.

3762, Failed to process the results from SQL statement.

Cause: Failed to process the results from the SQL statement.

Action: See secondary error.

3763, Failed to query the data files of the database.

Cause: Failed to query the data files from the database.

Action: See secondary error.

3764, Failed to query the log files used by the database.

Cause: Failed to query the log files used by the database.

Action: See secondary error.

3765, Failed to query table space information.

Cause: Failed to query tablespace information from the database.

Action: See secondary error.

B.2.5 OPMN Error Messages

The following are OPMN error messages.

3800, Failed trying to connect to OPMN Manager.

Cause: Error trying to connect to OPMN manager.

Action: Make sure OPMN manager is started.

3801, Failed trying to get topology information from OPMN Manager on {0}.

Cause: Error trying to get topology information from OPMN manager.

Action: Make sure OPMN manager is started and working correctly.

3802, Failed trying to stop OPMN Component {0}.

Cause: Failed trying to stop the specified OPMN component.

Action: See secondary error.

3803, Failed trying to start OPMN Component {0}.

Cause: Failed trying to start the specified OPMN component.

Action: See secondary error.

3900, Error creating Oracle database service because the service has already been marked for deletion. Please exit the Windows Service Control Manager on node {0}. Would you like to retry?

Cause: The user has the SCM open causing a service operation to fail.

Action: User must exit SCM GUI.

B.2.6 Net Services Error Messages

The following are Net Services error messages.

4000, Failed trying to get Net Services default domain for {0}.

Cause: Failed trying to get the Net Services default domain.

Action: See secondary error.

4001, Error trying to add net service name entry for {0}.

Cause: Failed trying to add the specified service name.

Action: See secondary error.

4002, Error trying to get net service name entry for {0}.

Cause: Failed trying to get the specified service name.

Action: See secondary error.

4003, Error trying to get host name from net service entry for {0}.

Cause: Failed trying to get the host name from the net service entry.

Action: See secondary error.

4004, Error trying to get host name from net service description.

Cause: Failed trying to get the host name from the net service description.

Action: See secondary error.

4005, Error trying to get net service listener information.

Cause: Failed trying to get the net service listener information.

Action: See secondary error.

4006, Error trying to create a net service default listener.

Cause: Failed trying to create a default listener.

Action: See secondary error.

4007, Error trying to add SID entry {0} to net service listener {1}.

Cause: Failed trying to add a SID entry to the listener.

Action: See secondary error.

4008, Error generating a command a script for the net service listener command: {0}.

Cause: Failed generating a command script for the listener.

Action: See secondary error.

4009, Error running the command script for the net service listener command: {0}.

Cause: Failed running the command script for the listener.

Action: See secondary error.

4010, Error adding the net service TNS entry for {0}.

Cause: Failed adding a TNS entry.

Action: See secondary error.

4011, Error trying to delete SID entry {0} to the net service listener {1}.

Cause: Failed trying to delete the SID entry to the listener.

Action: See secondary error.

4012, Error trying to save the listener configuration.

Cause: Listener information was modified and an attempt to save information failed.

Action: See secondary error.

4013, Error deleting net service TNS entry for {0}.

Cause: Failed deleting a TNS entry.

Action: See secondary error.

4014, Error starting the TNS listener using the lsnrctl command.

Cause: Failed to start the TNS listener.

Action: See secondary error.

4030, The command \"{0}\" failed due to timeout.

Cause: Command timed out.

Action: Increase timeout values in configuration file.

4031, Error getting environment variables using the env command.

Cause: The env command does not work.

Action: Make sure the `/bin` or `/usr/bin` directory contains the env executable.

4040, Error executing the external program or script.

Cause: The execution of the specified command failed.

Action: See secondary error.

4041, Failed to get the value of {0} from the TNS name descriptor {1}.

Cause: Failed to get the value for the given parameter from the TNS name descriptor.

Action: See secondary error.

4042, Failed to update the value of {0} for the TNS name descriptor {1}.

Cause: Failed to update the value of a given parameter in the TNS name descriptor.

Action: See secondary error.

4043, Failed to compare the TNS descriptor entry {0} with entry {1}.

Cause: Failed to compare the two TNS entries.

Action: See secondary error.

4044, Failed to generate a remote TNS name descriptor for the service name.

Cause: Failed to generate a remote TNS name descriptor for the given local database.

Action: See secondary error.

4045, Failed to get the remote TNS service name for the service name.

Cause: Failed to get the remote TNS service name for the given local database.

Action: See secondary error.

B.2.7 LDAP or OID Error Messages

The following are LDAP or OID error messages.

4101, Failed to connect to OID server on host {0}, port {1}.

Cause: Failed to the OID server on a given host and port.

Action: See secondary error.

4102, Failed to connect to OID server via SSL on host {0}, port {1}.

Cause: Failed to the OID server via SSL on a given host and port.

Action: See secondary error.

4103, User must specify host, port, user name, and password for the OID server.

Cause: User did not specify all the above parameters.

Action: User must specify all the above parameters in order to access the OID server.

4104, Failed to get the value of attribute {0} from OID server.

Cause: Failed to get the value of the given attribute.

Action: See secondary error.

4105, Failed to get the attributes for DN {0} from OID server

Cause: Failed to get the attributes of the given DN.

Action: See secondary error.

4106, Failed to get Oracle Application Server instances from OID server

Cause: Failed to get Oracle Application Server instances from the OID server.

Action: See secondary error.

4107, Failed to get infrastructure databases from OID server.

Cause: Failed to get infrastructure databases from the OID server.

Action: See secondary error.

4110, Cannot set current topology to file {0} because the file does not exist.

Cause: The topology file does not exist.

Action: Specify a filename of a file that exists.

4111, The current topology file \"{0}\" does not exist. Use the "set topology" command to specify a valid topology file.

Cause: The topology file does not exist.

Action: Specify a filename of a file that exists in dsa.conf.

B.2.8 System Error Messages

The following are system error messages.

4900, An exception occurred on the server.

Cause: A server exception occurred.

Action: See secondary error.

4901, A null pointer exception occurred on the server.

Cause: Software error.

Action: See secondary error.

4902, Object not found in clipboard for key {0}.

Cause: Software error.

Action: See secondary error.

4903, The minimum succeed value of {0} was not met for the workers in group {1}.

Cause: A group of workers belonging to the same group requires that a minimum number of them succeed. That minimum succeed value was not met.

Action: See secondary error.

4950, An error occurred on host {0} with IP {1} and port {2}.

Cause: An error occurred on the server.

Action: See secondary error.

B.2.9 Warning Error Messages

The following are warning error messages.

15305, Warning: Problem gathering summary information for backup.

Cause: Error during the gatherInfo step of the backup topology operation.

Action: Check secondary error.

15306, Warning during undo processing.

Cause: Error occurred during undo processing.

Action: Check secondary error.

B.2.10 OracleAS Database Error Messages

The following are OracleAS database error messages.

15604, Error finishing up creating the physical standby database.

Cause: Failed to finish creating the standby database.

Action: See secondary error.

15605, Error creating the physical standby database.

Cause: Failed to the create the standby database.

Action: See secondary error.

15606, Failed to perform a sync database operation on the primary topology.

Cause: Failed to perform a sync database operation on the primary topology.

Action: See secondary error.

15607, Failed to perform a sync database operation on the standby topology.

Cause: Failed to perform a sync database operation on the standby topology.

Action: See secondary error and log file for more information.

15608, Invalid backup mode specified in the template file {0}.

Cause: Error trying to process a backup mode value in the template file.

Action: Correct backup mode and retry the operation.

15609, Failed to get database backup files.

Cause: Error trying to get the database backup files.

Action: See secondary error.

B.2.11 OracleAS Topology Error Messages

The following are OracleAS topology error messages.

15620, An Invalid Topology was specified.

Cause: Error trying to process a topology object.

Action: Retrieve a valid topology object.

15621, Error trying to verify topology {0}.

Cause: The specified topology had an error during the verify operation.

Action: See secondary error for more information.

15622, Error trying to verify instance {0}.

Cause: The specified instance had an error during the verify operation.

Action: See secondary error for more information.

15623, Topology {0} is not symmetrical with topology {1}.

Cause: The specified topologies are not symmetrical.

Action: See secondary error for more information.

15624, An Invalid Topology was specified. Topology {0} does not contain any valid instances.

Cause: Error trying to process a topology object. Topology object did not contain a valid instance.

Action: Retrieve a valid topology object with at least one instance.

15625, Could not find matching instance {0} in Topology {1}.

Cause: Could not get matching instances. Topologies do not appear to be symmetrical.

Action: Make the topologies symmetrical.

15626, Topologies are not symmetrical because topology name {0} is not the same as topology name {1}.

Cause: Topology names are not the same and therefore topologies are not symmetrical.

Action: Make the topologies symmetrical.

15627, Instance {0} is not symmetrical because of different Oracle Home names {1}.

Cause: Instance Home names are not symmetrical in the specified topologies.

Action: Make the topologies symmetrical.

15628, Instance {0} is not symmetrical because of different Oracle Home paths {1}.

Cause: Instance Home paths are not symmetrical in the specified topologies.

Action: Make the topologies symmetrical.

15629, Instance {0} is not symmetrical, because of different host names {1}, {2}.

Cause: Instance host names are not symmetrical in the specified topologies.

Action: Make the topologies symmetrical.

15630, The specified instance {0} could not be found.

Cause: The specified instance information could not be found on this node.

Action: Either the wrong instance name or host name was specified on the request to the server.

15631, The primary and standby topologies appear to be identical because both have instance {0} on host {1}.

Cause: An instance can only be in a member of one topology, it appears that the primary and standby topologies are the same.

Action: Specify a primary and separate standby topology.

15632, The Home that contains instance {0} could not be found.

Cause: The specified instance could not be found in any Home on this node.

Action: The Oracle home information on the system is incorrect.

15633, An Invalid Topology was specified. Topology contains a duplicate instance named {0}.

Cause: Topology information obtained from OPMN contains a duplicate instance.

Action: Check OPMN to ensure that the topology information listed is correct.

B.2.12 OracleAS Backup and Restore Error Messages

The following are OracleAS backup and restore error messages.

15681, Must specify a backup directory.

Cause: A backup directory must be specified for the operation to complete successfully.

Action: Check secondary error.

15682, Failed to initialize configure file: {0}.

Cause: Failed to initialize the configure file for backup script.

Action: Check secondary error.

15683, The ha directory does not exist in Oracle Home {0}.

Cause: The ha directory does not exist in the OracleAS Oracle home.

Action: Make sure the ha directory which contains the backup and restore scripts is copied to the OracleAS Oracle home.

15684, Failed to generate the configuration file for the backup and restore script.

Cause: Failed to generate the configure file for the backup and restore script.

Action: Check secondary error.

15685, Failed to backup configuration data for instance {0}.

Cause: Failed to backup configuration data for the specified instance.

Action: Check secondary error.

15686, Failed to restore configuration data for instance {0}.

Cause: Failed to restore configuration data for the specified instance.

Action: Check secondary error.

15687, Failed to get the database backup files.

Cause: Failed to get the database backup file names from the log.

Action: Check secondary error.

15688, Error running the config script.

Cause: Failed to run the config script.

Action: Check the log file generated by the config script.

15689, Error running the backup script.

Cause: Failed to run the backup script.

Action: Check the log file generated by the backup script.

15690, Error running the restore script.

Cause: Failed to run the restore script.

Action: Check the log file generated by the restore script.

15691, No zip file was found.

Cause: No zip file was found.

Action: Make sure a successful backup has been performed.

15692, The config file {0} is empty.

Cause: The specified configure file is empty.

Action: Copy the original configure file from the "ha" directory where backup restore scripts are located.

15693, No zip file was specified.

Cause: User did not specify a zip file for the unzip operation.

Action: Internal error.

15694, Error executing step - {0} of Backup topology.

Cause: Backup topology failed at the specified step.

Action: Check secondary error.

15695, Error executing step - {0} of Restore topology.

Cause: Restore topology failed at the specified step.

Action: Check secondary error.

15696, Error initializing backup topology operation.

Cause: Error initializing backup topology operation.

Action: Check secondary error.

15697, Error during backup topology operation - backup step.

Cause: Error during backup step processing of backup topology.

Action: Check secondary error.

15698, Error during backup topology operation - copy step.

Cause: Error during copy step processing of backup topology.

Action: Check secondary error.

15699, Error initializing restore topology operation.

Cause: Error initializing restore topology operation.

Action: Check secondary error.

15700, No backup file was found.

Cause: No backup file was found.

Action: Make sure a successful backup has been performed.

15701, Failed to restore configuration with the DCM-resyncforce option for instance {0}.

Cause: Failed to restore configuration with the DCM-resyncforce option.

Action: Check secondary error.

15702, Error initializing the clone instance operation.

Cause: Error initializing the clone instance operation.

Action: Check the secondary error.

15703, Error initializing the clone topology operation.

Cause: Error initializing the clone home operation.

Action: Check the secondary error.

15704, Error: Oracle home of the instance to be cloned {0} already exists.

Cause: Error cloning instance, the Oracle home already exists

Action: Clean up the Oracle home and retry.

15705, cloning instance {0}. Cloning requires OPMN to be stopped, therefore the OracleAS Guard server must be started using asgctl .

Cause: Cloning requires that OPMN be stopped, which will cause the OracleAS Guard server (DSA server process) to be stopped. This will cause the clone to fail.

Action: Use opmnctl to stop the OracleAS Guard server (DSA server process). Then use the asgctl startup topology command to restart OracleAS Guard server for this instance.

15706, Stop the backup home image operation in response to the user's request.

Cause: Stop the backup home operation because the user entered NO.

Action: None.

15707, Stop the restore home image operation in response to the user's request.

Cause: Stop the restore home operation because the user entered NO.

Action: None.

B.2.13 OracleAS Guard Synchronize Error Messages

The following are OracleAS Guard synchronize error messages.

15721, Failed to initialize a DUF database object.

Cause: Failed to initialize a DufDb object.

Action: Check secondary error.

15722, No topology information is available to perform the topology operation.

Cause: No topology information is available to perform the topology operation.

Action: Check secondary error.

15723, No instances are found in the topology's backup list.

Cause: The topology's backup list is empty.

Action: Check secondary error.

15724, Failed to get the standby host list.

Cause: Failed to get the standby host list.

Action: Check secondary error.

15725, Failed to backup OracleAS configuration data for topology {0}.

Cause: Failed to backup OracleAS topology configuration data.

Action: Check secondary error.

15726, Failed to restore OracleAS configuration data for topology.

Cause: Failed to restore OracleAS topology configuration data.

Action: Check secondary error.

15727, Failed to backup OracleAS infrastructure database {0}.

Cause: Failed to backup OracleAS topology infrastructure database.

Action: Check secondary error.

15728, Failed to restore OracleAS infrastructure database {0}.

Cause: Failed to restore OracleAS topology infrastructure database.

Action: Check secondary error.

15729, Failed to perform the sync topology operation.

Cause: Failed to perform the sync topology operation.

Action: Check secondary error.

B.2.14 OracleAS Guard Instantiate Error Messages

The following are OracleAS Guard instantiate error messages.

15751, Error executing step {0} of instantiate topology operation.

Cause: The instantiate topology operation failed at the specified step.

Action: Check secondary error.

15752, Failed to load remote topology information.

Cause: Failed to load remote topology information.

Action: Make sure the user specified the correct host name for the topology and that the OPMN processes are running on the topologies.

15753, Error preparing to instantiate topology on host {0}.

Cause: Error preparing to instantiate topology.

Action: Check secondary error.

15754, Error instantiating database {0}.

Cause: Error instantiating the database.

Action: Check secondary error.

15755, Error finishing up instantiating database {0}.

Cause: Error finishing up instantiating the database.

Action: Check secondary error.

15756, Error initializing instantiate topology operation.

Cause: Error initializing the instantiate topology operation.

Action: Check secondary error.

15757, Error initializing switchover topology operation.

Cause: Error initializing the switchover topology operation.

Action: Check secondary error.

15770, The instance {0} specified in the topology file does not match the instance {1} in home {2}.

Cause: The topology file is incorrect.

Action: Run the "discover topology command" from asgctl.

15771, The topology file {0} is the wrong version Please delete the file and rediscover the topology.

Cause: The topology file is incorrect.

Action: Run the "discover topology command" from asgctl.

15772, The topology file {0} does not contain an entry for the discovery host {1}.

Cause: The topology file is incorrect.

Action: Run the "discover topology command" from asgctl.

15773, The standby topology does not contain a entry for the mandatory primary instance {0}.

Cause: The topology file is incorrect.

Action: Run the "discover topology command" from asgctl.

15774, The host name {0} in the standby topology net descriptor for database {1} resolves to a primary host address {2}.

Cause: The topology file is incorrect.

Action: Run the "discover topology command" from asgctl.

15775, The standby topology host name {0} for the instance {1} resolves to a primary host address.

Cause: The topology file is incorrect.

Action: Run the "discover topology command" from asgctl.

15776, Error accessing the OID server.

Cause: Unable to access the OID server.

Action: Specify the correct OID information and make sure the OID server is running.

15777, Error: OID information needed to access the server was not specified.

Cause: Unable to access the OID server.

Action: Specify the correct OID information.

15778, Error getting database information for SID {0} from host {1}. This instance will be excluded from the topology.xml file.

Cause: Unable to get database information for the topology database.

Action: None.

15779, Error getting instance information for instance {0} from host {1}. This instance will be excluded from the topology.xml file.

Cause: Unable to get information for an instance.

Action: None.

15780, Instance {0} cannot be found in the topology.

Cause: The instance name does not exist in the topology file.

Action: Perform the asgctl discover topology command.

Index

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W