Oracle® Fusion Middleware

High Availability Guide

12*c* (12.2.1.3.0)
E95492-04
October 2023

Oracle Fusion Middleware High Availability Guide, 12*c* (12.2.1.3.0)

E95492-04

Primary Author: Oracle Corporation

Contents

9 Administration Server High Availability

Part III Component Procedures

10 Configuring High Availability for Oracle Identity Governance Components

11 Configuring High Availability for Oracle Access Manager Components

12 Configuring High Availability for Oracle Directory Services Components

13 Configuring High Availability for Web Tier Components

14 Configuring High Availability for SOA Components

15 Configuring High Availability for Oracle WebCenter Components

16 Configuring High Availability for Other Components

Preface

This preface contains these sections:

Audience

The document is intended for administrators, developers, and others whose role is to deploy and manage Oracle Fusion Middleware with high availability requirements.

Purpose of this Guide

The purpose of this guide is to serve as a reference document to set up a highly available environment. Use this guide in conjunction with your product's installation and administration guides.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Documents

Refer to the Oracle Fusion Middleware Library for additional information.

- See About Key Oracle Fusion Middleware Concepts in *Understanding Oracle Fusion Middleware* for information on the common terms and concepts in an Oracle Fusion Middleware environment.

- See Getting Started Managing Oracle Fusion Middleware in *Administering Oracle Fusion Middleware* for information on managing your Oracle Fusion Middleware environment after installation and configuration is complete.
- *Oracle Fusion Middleware Tuning Performance Guide*
- *Oracle Fusion Middleware Administering Clusters for Oracle WebLogic Server*
- For release-related information, see Fusion Middleware Release Notes.

For definitions of unfamiliar terms found in this and other books, see the Glossary.

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.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Part I
Introduction to High Availability

Part I contains the following topics:

1

Introduction and Roadmap

Read this section for information on how and why to use this document and high availability environments.

- How to Use This Document
  Use this document to reference information on high availability concepts and tasks as you set up a highly available environment.
- Setting up a Highly Available Environment
  A highly available environment requires preparation and planning before you configure it.
- New and Changed Features in This Release
  Oracle Fusion Middleware 12*c* (12.2.1.3.0) includes new and changed concepts and features.
- What is High Availability?
  High availability is the ability of a system or device to be available when it is needed.
- High Availability Solutions
  You can categorize high availability solutions into local **high availability solutions** that provide high availability in a single data center deployment, and **disaster recovery solutions**.
- About the Oracle Fusion Middleware Standard HA Topology
  Oracle recommends a standard high availability topology that has elements that this topic describes.

How to Use This Document

Use this document to reference information on high availability concepts and tasks as you set up a highly available environment.

Before you use this document, you must have a standard installation topology (SIT) set up for your product. This is the required starting point for setting up high availability. See the topics Understanding the Oracle Fusion Middleware Infrastructure Standard Installation Topology and Roadmap for Installing and Configuring the Standard Installation Topology to set up the SIT.

Setting up a Highly Available Environment

A highly available environment requires preparation and planning before you configure it.

Table 1-1 describes tasks to set up a highly available environment and additional resources for information.

Table 1-1 Setting up a Highly Available Environment

Task	Description	For more information
Performing administrative tasks and preparing your environment	Common tasks to perform on a newly-created domain.	See Administering and Preparing your WebLogic Domain for High Availability in your product installation guide.
Planning your WebLogic Server Installation	Covers understanding your topology and determining the distribution, components, and features you need.	See Preparing for an Oracle Fusion Middleware Installation in *Planning an Installation of Oracle Fusion Middleware*.
Installing the WebLogic Server Software	Describes how to start the installation process and go through installation screens.	See Installing the Oracle Fusion Middleware Infrastructure Software in your product installation guide.
Configuring a domain	Creating and configuring a domain	See Configuring your Oracle Fusion Middleware Infrastructure Domain in your product installation guide.
Managing Oracle Fusion Middleware	Includes how to: start and stop, change ports, deploy applications, and back up and recover Oracle Fusion Middleware.	See *Oracle Fusion Middleware Administrator's Guide*.
Monitoring and optimizing performance in the Oracle Fusion Middleware environment.	For components that affect performance, use multiple components for optimal performance, and design applications for performance.	See *Oracle Fusion Middleware Tuning Performance Guide*.
Setting up a product-specific enterprise deployment	Oracle best practices blueprints based on proven Oracle high availability and security technologies and recommendations for a product-specific enterprise deployment.	See your product's Enterprise Deployment Guide.
Administering the product environment	To deploy, manage, monitor, and configure applications using the product.	See your product's Administrator's Guide.
Configuring Node Manager	Use Node Manager to start, shut down, and restart the Administration Server and Managed Servers from a remote location. It is an essential tool for a high availability environment.	See *Administering Node Manager for Oracle WebLogic Server*.

New and Changed Features in This Release

Oracle Fusion Middleware 12*c* (12.2.1.3.0) includes new and changed concepts and features.

- Support for WebCenter. See Configuring High Availability for Oracle WebCenter Components .

- Support for BI. See Deploying BI.

> **Note:**
>
> For a comprehensive list of new and deprecated:
>
> - **WebLogic Server features** in this release, see *Oracle Fusion Middleware What's New in Oracle WebLogic Server.*
> - **Terms** in this release, see New and Deprecated Terminology for 12c in *Understanding Oracle Fusion Middleware Concepts*.

What is High Availability?

High availability is the ability of a system or device to be available when it is needed.

A high availability architecture ensures that users can access a system without loss of service. Deploying a high availability system minimizes the time when the system is down, or unavailable, and maximizes the time when it is running, or available.

High availability comes from redundant systems and components. You can categorize high availability solutions by their level of redundancy into **active-active** solutions and **active-passive** solutions.

- Active-Active High Availability Solutions
  An **active-active solution** deploys two or more active servers to improve scalability and provide high availability.
- Active-Passive High Availability Solutions
  An **active-passive solution** deploys one active instance that handles requests and one passive instance that is on standby.

Active-Active High Availability Solutions

An **active-active solution** deploys two or more active servers to improve scalability and provide high availability.

In active-active deployments, all instances handle requests concurrently. Oracle recommends active-active solutions for all single-site middleware deployments.

Active-Passive High Availability Solutions

An **active-passive solution** deploys one active instance that handles requests and one passive instance that is on standby.

If the active node fails, the standby node activates and the middle-tier components continue servicing clients from that node. All middle-tier components fail over to the new active node. No middle-tier components run on a failed node after failover.

Oracle supports active-passive deployments for all components.

High Availability Solutions

You can categorize high availability solutions into local **high availability solutions** that provide high availability in a single data center deployment, and **disaster recovery solutions**.

- Local high availability solutions can protect against process, node, and media failures, as well as human errors, ensuring availability in a single data center deployment.
- Disaster recovery solutions are usually geographically distributed deployments that protect applications from disasters such as floods or regional network outages. You can protect against physical disasters that affect an entire data center by deploying geographically-distributed disaster recovery solutions. For more on disaster recovery for Oracle Fusion Middleware components, see Overview of Disaster Recovery in *Oracle Fusion Middleware Disaster Recovery Guide*.

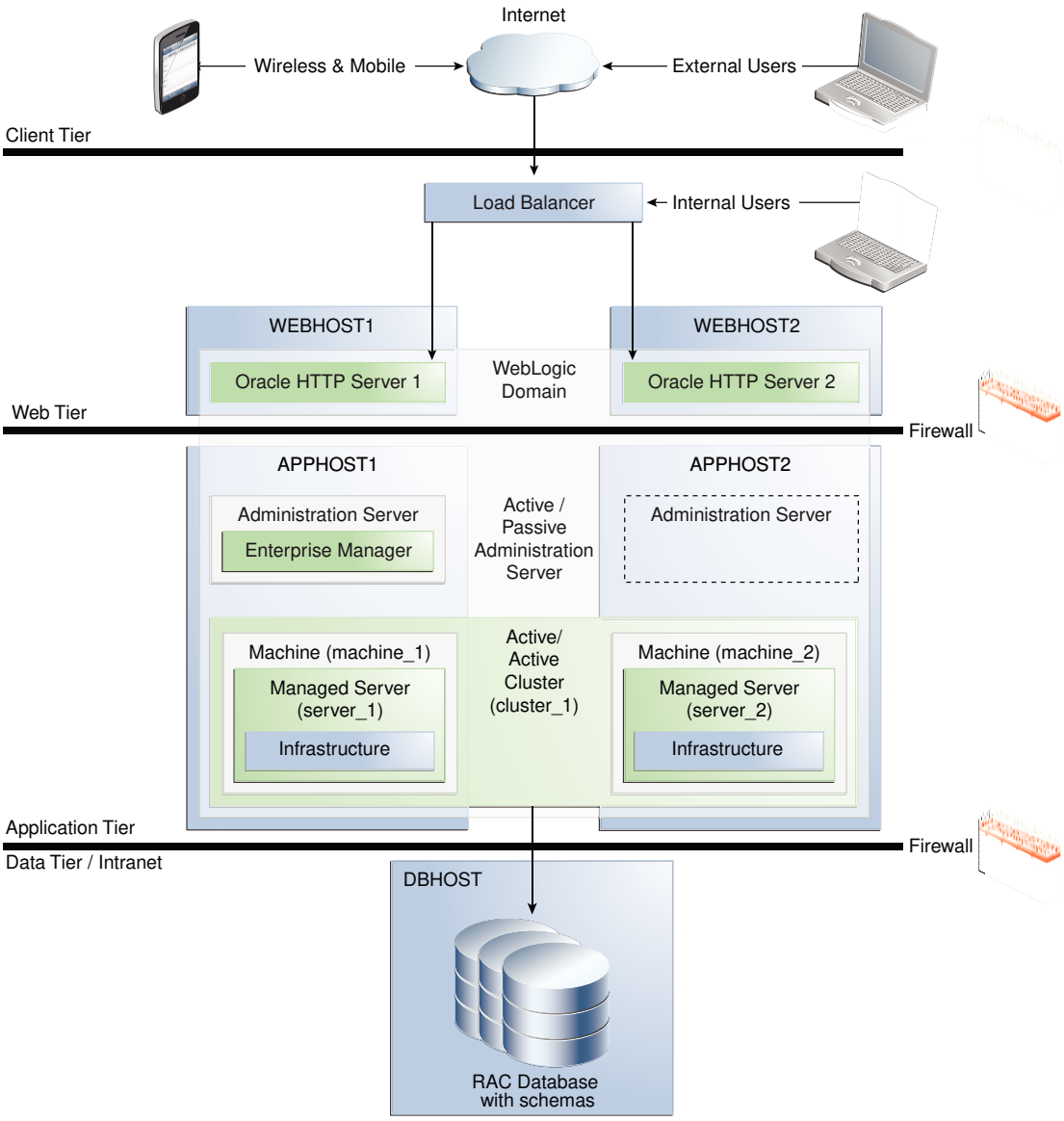
About the Oracle Fusion Middleware Standard HA Topology

Oracle recommends a standard high availability topology that has elements that this topic describes.

Figure 1-1 shows the recommended standard high availability topology for a local, highly available Oracle Fusion Middleware deployment.

This deployment is consistent with the infrastructure SIT and Oracle HTTP Server SIT if you followed steps in Roadmap for Installing and Configuring the Standard Installation Topology and Roadmap for Installing and Configuring Oracle HTTP Server in a WebLogic Server Domain.

Figure 1-1 Oracle Fusion Middleware Highly Available Deployment Topology (Typical Enterprise)



This topology shows a multi-tiered architecture. Users access the system from the client tier. Requests go through a hardware load balancer, which routes them to Web servers running Oracle HTTP Servers in the web tier. Web servers use Proxy Plug-in (`mod_wl_ohs`) to route requests to the WebLogic cluster in the application tier. Applications running on the WebLogic cluster in the application tier then interact with the database cluster in the data tier to service the request.

- Elements in the Standard High Availability Topology
  A Standard High Availability Installation Topology includes certain elements.

Elements in the Standard High Availability Topology

A Standard High Availability Installation Topology includes certain elements.

Table 1-2 describes elements in Figure 1-1.

Table 1-2 Description of Elements in the Oracle Fusion Middleware Infrastructure Standard High Availability Topology

Element	Description and Links to Additional Documentation
APPHOST	Machine that hosts the application tier.
WEBHOST	Machine that hosts the web tier.
WebLogic Domain	A logically related group of Java components, in this case, the Administration Server, Managed Servers, and other software components. For more, see What is an Oracle WebLogic Server Domain? in *Understanding Oracle Fusion Middleware.*
Administration Server	Central control entity of a domain. Maintains a domain's configuration objects and distributes configuration changes to Managed Servers.
Enterprise Manager	Oracle Enterprise Manager Fusion Middleware Control. The main tool that you use to manage a domain.
Cluster	A collection of multiple WebLogic Server instances running simultaneously and working together.
Machine	Logical representation of the computer that hosts one or more WebLogic Server instances (servers). Machines are the logical 'glue' between Managed Servers and Node Manager; to start or stop a Managed Server with Node Manager, the Managed Server must be associated with a machine.
Managed Server	Host for applications, application components, Web services, and their associated resources. See Oracle Enterprise Manager Fusion Middleware Control in *Understanding Oracle Fusion Middleware*.
Infrastructure	Collection of services that includes: • Metadata repository (MDS). Contains metadata for components, such as Oracle Application Developer Framework. See What is the Metadata Repository? in *Understanding Oracle Fusion Middleware*. • Oracle Application Developer Framework (Oracle ADF) • Oracle Web Services Manager (OWSM)

Note:

- To view a figure of the Infrastructure SIT and follow a roadmap to install it, see Understanding the Infrastructure Standard Installation Topology in *Oracle Fusion Middleware Installing and Configuring the Oracle Fusion Middleware Infrastructure*.
- To view a figure of the Oracle HTTP Server SIT and follow a roadmap to install it, see Introducing the Oracle HTTP Server Standard Installation Topologies in *Installing and Configuring Oracle HTTP Server*.

2

High Availability Concepts

High availability involves elements such as load balancing, failover, Real Application Clusters, and profiles (settings).

- Oracle Fusion Middleware Concepts
  Get familiar with important concepts before scaling out.
- Server Load Balancing in a High Availability Environment
  Typically, a load balancer front ends high availability deployments.
- Application Failover
  There are different types of failover and application behavior.
- Real Application Clusters
  Oracle Real Application Clusters (RAC) enable you to cluster an Oracle database. A **cluster** comprises multiple interconnected computers or servers that appear as if they are one server to end users and applications.
- Coherence Clusters and High Availability
  Every standard installation topology (SIT) includes a standard Coherence cluster. This cluster is a starting point for additional configuration.
- Disaster Recovery
  For maximum availability, you may need to deploy services at different geographical locations to protect against entire site failures due to unforeseen disasters and natural calamities.
- Install Time Configuration (Profiles)
  There are domain, file, and database-based profile types to consider as you configure a SIT.
- Application and Service Failover for JMS and JTA
  Migration in WebLogic Server is the process of moving 1) a clustered WebLogic Server instance or 2) a component running on a clustered instance elsewhere if failure occurs.
- Roadmap for Setting Up a High Availability Topology
  Oracle recommends completing install and configuration steps in a certain order to set up an example high availability topology.

Oracle Fusion Middleware Concepts

Get familiar with important concepts before scaling out.

Table 2-1 describes relevant topics in *Oracle Fusion Middleware Understanding Oracle Fusion Middleware Concepts*.

Table 2-1 Oracle Fusion Middleware Concepts

For information on...	See this topic...
Oracle Home, Oracle Common, WebLogic Server Domain	What Are the Key Oracle Fusion Middleware Directories?

Table 2-1 (Cont.) Oracle Fusion Middleware Concepts

For information on...	See this topic...
WebLogic Server Domain	What Is an Oracle WebLogic Server Domain?
Administration Server	What Is the Administration Server?
Managed Servers and Clusters	Understanding Managed Servers and Managed Server Clusters
Node Manager	What Is Node Manager?

> **Note:**
>
> See Communications in a Cluster in *Oracle Fusion Middleware Administering Clusters for Oracle WebLogic Server*

Server Load Balancing in a High Availability Environment

Typically, a load balancer front ends high availability deployments.

- About Load Balancing
  Load balancing is the even distribution of jobs and associated communications across computing and networking resources in your environment.
- Third-Party Load Balancer Requirements
  You can use third-party load balancers in your high availability deployments, as long as they have certain features.
- Configuring Third-Party Load Balancers
  The detailed load balancer configuration steps depend on two elements that are explained in this topic.
- Server Load Balancing with Oracle HTTP Server or Oracle Traffic Director
  Oracle has two options for server load balancing products: Oracle HTTP Server and Oracle Traffic Directory.

About Load Balancing

Load balancing is the even distribution of jobs and associated communications across computing and networking resources in your environment.

You use Oracle HTTP Server or Oracle Traffic Director to configure load balancing between different components or applications. See Server Load Balancing with Oracle HTTP Server or Oracle Traffic Director.

You can also combine of a load balancer and Oracle HTTP Server (as Figure 1-1 shows) to provide maximum availability.

Third-Party Load Balancer Requirements

You can use third-party load balancers in your high availability deployments, as long as they have certain features.

Oracle recommends load balancers that support *sticky* session routing. **Sticky session routing** is the ability, for the entire session, to route traffic to the same server that processes the first request.

External load balancer must have these features:

- Ability to load-balance traffic to a pool of real servers through a virtual host name: clients access services using the *virtual* host name instead of *actual* host names. The load balancer can then load balance requests to servers in the pool. Typically, the load balancer can balance across Oracle HTTP Server instances and then the Oracle HTTP Servers can balance across application servers.
- Port translation configuration.
- Monitoring of ports (HTTP and HTTPS).
- Ability to configure virtual server names and ports on the load balancer.
- 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 clusters, you must configure the load balancer with a virtual server and ports for HTTP and HTTPS traffic.
- 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 virtual server names.
- Resource monitoring/port monitoring/process failure detection: the load balancer must be able to detect service and node failures and to stop directing non-Oracle Net traffic to a failed node. If your external load balancer can automatically detect failures, Oracle recommends that you use it.
- Fault tolerant mode: Oracle recommends that you configure the load balancer to be in fault-tolerant mode.
- Virtual server returning to calling client: Oracle highly recommends that you configure the load balancer virtual server to return immediately to the calling client when back-end services that it forwards traffic to are unavailable. This is preferred over a client disconnecting on its own after a timeout based on TCP/IP settings on a client machine.
- SSL acceleration. Oracle recommends this feature but doesn't require it.
- Client IP Address (Preserving): the load balancer must be able to insert a request's original client IP address in an X-Forwarded-For HTTP header to preserve it.

Configuring Third-Party Load Balancers

The detailed load balancer configuration steps depend on two elements that are explained in this topic.

- The environment you are using the load balancer in.
- The type of load balancer you are using.

For these reasons, Oracle recommends that you follow your load balancer's documentation. For high-level load balancer configuration steps, see the enterprise deployment guide for the component you are working with.

Server Load Balancing with Oracle HTTP Server or Oracle Traffic Director

Oracle has two options for server load balancing products: Oracle HTTP Server and Oracle Traffic Directory.

Table 2-2 Server Load Balancing Products

Product	Description
Oracle HTTP Server (OHS)	Web server with a built-in WebLogic Server Proxy Plug-In module to act as HTTP front-end for WebLogic servers. Receives incoming requests from clients and load balances each request to WebLogic servers. The OHS `mod_wl_ohs` module routes requests to WebLogic servers. Has the same load balancing functionality as `mod_weblogic` (Oracle WebLogic Server Plug-in for Apache HTTP Server). See Configuring the mod_wl_ohs Plug-In for Oracle HTTP Server in *Oracle Fusion Middleware Using Web Server 1.1 Plug-Ins with Oracle WebLogic Server*. See Configuring mod_wl_ohs.conf for more on the `mod_wl_ohs` module.
Oracle Traffic Director	Fast, reliable, and scalable layer-7 software load balancer. Reliable entry point for all HTTP, HTTPS and TCP traffic. Distributes client requests based on specified load-balancing method, routes requests based on specified rules, caches frequently accessed data, prioritizes traffic, and controls service quality. See Features of Oracle Traffic Director.

Application Failover

There are different types of failover and application behavior.

- About Failover, Application Failover, and State
  Failover is relocating an overloaded or failed resource such as a server, disk drive, or network to its backup location.
- Session Failover Requirements
  For seamless failover, an application must meet certain conditions.
- Application Failover Expected Behavior
  If you configure the environment correctly, users don't notice when an application instance in a cluster becomes unavailable.

About Failover, Application Failover, and State

Failover is relocating an overloaded or failed resource such as a server, disk drive, or network to its backup location.

Application failover is when an application component doing a certain job becomes unavailable and a copy of the failed component finishes the job.

State is information about what has been done on a job. WebLogic Server maintains state information using session replication and replica-aware stubs. If a component

unexpectedly stops doing its job, replication techniques enable a copy of the component to pick up where the failed component stopped and finish the job.

Session Failover Requirements

For seamless failover, an application must meet certain conditions.

> **Note:**
>
> Oracle applications meet these session failover requirements unless a specific exception is made for an application.

An application must meet these conditions to failover seamlessly:

- The application is in a cluster and at least one member of the cluster is available to serve the request.
- For stateful applications, state replication is configured correctly.
- If you use Oracle HTTP Server, the server is configured with the WebLogic Cluster directive to balance among all available application instances.
- If you are using a hardware load balancer, the load balancer is:
  - Routing traffic to all available instances
  - Configured correctly with a health monitor to mark unavailable instances
  - Configured to support persistence of session state

Application Failover Expected Behavior

If you configure the environment correctly, users don't notice when an application instance in a cluster becomes unavailable.

The application failover sequence of events is (for example):

1. A user makes a request. A hardware load balancer routes it to Instance A of the application.
2. Instance A of the application becomes unavailable due to node failure, process failure, or network failure.
3. The hardware load balancer marks Instance A as unavailable.
4. The user makes another request. The request is routed to Instance B.
5. Instance B is configured as a replication partner of Instance A and has the user's session state.
6. The application resumes using the session state on Instance B. The user continues working without interruption.

> **Note:**
>
> See *Domain Template Reference* for domain and extension templates that support high availability.
>
> See Failover and Replication in a Cluster in *Administering Clusters for Oracle WebLogic Server* for more on failover and replication at the application level.

Real Application Clusters

Oracle Real Application Clusters (RAC) enable you to cluster an Oracle database. A **cluster** comprises multiple interconnected computers or servers that appear as if they are one server to end users and applications.

Oracle RAC uses Oracle Clusterware for the infrastructure to bind multiple servers so that they operate as one system. Along with a collection of hardware (cluster), Oracle RAC unites the processing power of each component to become a single, robust computing environment. Oracle RAC provides a highly scalable and highly available database for Oracle Fusion Middleware.

Every Oracle RAC instance in a cluster has equal access and authority. Node and instance failure may affect performance but don't result in downtime because the database service is available or can be made available on surviving server instances.

> **Note:**
>
> For more on Oracle RAC see:
>
> - Introduction and Roadmap in *Oracle Fusion Middleware Administering Clusters for Oracle WebLogic Server*
> - Overview of Oracle Clusterware in *Oracle Clusterware Administration and Deployment Guide*
> - Overview of Oracle RAC in *Oracle Real Application Clusters Administration and Deployment Guide*

Coherence Clusters and High Availability

Every standard installation topology (SIT) includes a standard Coherence cluster. This cluster is a starting point for additional configuration.

A **Coherence cluster** is a collection of Java Virtual Machine (JVM) processes running Coherence. In 12c, these processes are called **WebLogic Managed Coherence Servers**. JVMs that join a cluster are **cluster members** or **cluster nodes**. Cluster members can be:

- Dedicated storage members
- Client members that have storage disabled
- Proxy members that allow non-cluster members to access Coherence caches

Cluster members communicate using Tangosol Cluster Management Protocol (TCMP). Cluster members use TCMP for both multicast communication (broadcast) and unicast communication (point-to-point communication).

Coherence characteristics include the following:

- Each domain typically contains one Coherence Cluster.
- Each managed Coherence server in a domain is associated with a Coherence cluster, defined through a Coherence Cluster System Resource.
- Each application has its Coherence configuration in a Grid Archive (GAR) file, which deploys with the application and to all dedicated storage nodes.

All applications that use Coherence use the cluster associated with the managed Coherence server and deploy their GAR files co-located with their applications. Table 2-3 lists sources of information about Coherence.

Table 2-3 Coherence and Coherence Clusters

For information on...	See this topic...
Coherence concepts and features	Introduction to Coherence in *Oracle Fusion Middleware Developing Applications with Oracle Coherence*
Creating Coherence clusters	Setting Up a WebLogic Server Domain Topology for Coherence in *Coherence Administrator's Guide*
Configuring a Coherence Cluster	Configuring and Managing Coherence Clusters in *Administering Clusters for Oracle WebLogic Server*

Disaster Recovery

For maximum availability, you may need to deploy services at different geographical locations to protect against entire site failures due to unforeseen disasters and natural calamities.

Oracle Fusion Middleware products support the configuration of a geographically separate standby site to act as a backup. Applications and services can fail over to this backup in the event of natural or unplanned outages at a production site.

For more on disaster recovery, see Introduction to Oracle Fusion Middleware Disaster Recovery.

Install Time Configuration (Profiles)

There are domain, file, and database-based profile types to consider as you configure a SIT.

- Domain (Topology) Profiles
  Use the Configuration Wizard or WebLogic Scripting Tool (offline) to set up domains.
- Persistence Profile Types
  Persistence profiles are a collection of settings that Oracle recommends for a specific persistence environment. There are two primary persistence profiles: database and file based.
- Post-Configuration Defaults
  When you complete a standard installation, a domain is set up with a file-based persistence profile.

Domain (Topology) Profiles

Use the Configuration Wizard or WebLogic Scripting Tool (offline) to set up domains.

See Creating a WebLogic Domain in *Creating WebLogic Domains Using the Configuration Wizard* to create, update, and configure domains.

12*c* (12.2.1.3.0) installation guides have steps to set up a single machine, multi-server domain, which is the **standard installation topology**. About the Oracle Fusion Middleware Standard HA Topology describes this topology in detail. See:

- About the Oracle HTTP Server Installation in *Oracle Fusion Middleware Installing and Configuring Oracle HTTP Server*
- Configuring the Oracle Fusion Middleware Infrastructure Domain in *Oracle Fusion Middleware Installing and Configuring the Oracle Fusion Middleware Infrastructure*.

Persistence Profile Types

Persistence profiles are a collection of settings that Oracle recommends for a specific persistence environment. There are two primary persistence profiles: database and file based.

Table 2-4 shows persistence types for both profiles.

Although you can *mix & match* a component or service with the persistence profile, the persistence type groups work together optimally. Oracle recommends that you use all options consistently within their respective profile.

Table 2-4 Persistence Types for Database and File Persistence Profiles

Component/Service	Database Persistence Profile	File Persistence Profile
JMS	WLS JMS in Database Store	WebLogic Server JMS in File Store
JTA	JTA in Database Store	JTA in File Store
OPSS	Database Store	Database Store
MDS	Database Store	Database Store
Service Table	Database Store	Database Store
Failover	Whole Server Migration	Whole Server Migration

> **Note:**
>
> An MDS data source has a WebLogic Server file persistence store allocated along with the data source. Because you use the file persistence store only in development mode, you can ignore it for high availability purposes. You don't need to recover the file persistence store in the event of failure.

> **Note:**
>
> See Interoperability with Supported Databases in the *Interoperability and Compatibility Guide* for database version requirements for selected products and features.

Post-Configuration Defaults

When you complete a standard installation, a domain is set up with a file-based persistence profile.

To configure database-based persistence for JMS/JTA resources, see JMS and JTA High Availability. To set up whole server migration, see Whole Server Migration .

Table 2-5 describes additional sources of information.

> **Note:**
>
> Some products may have specific requirements for shared file stores; Oracle recommends that you refer to your product's requirements for details.

Table 2-5 Domain Configuration Topics

For additional information on...	See this topic...
Shared file systems to use with a file persistence profile	Using Shared Storage
JMS and JTA	About Migratable Targets for JMS and JTA Services
Failover	Application Failover

Application and Service Failover for JMS and JTA

Migration in WebLogic Server is the process of moving 1) a clustered WebLogic Server instance or 2) a component running on a clustered instance elsewhere if failure occurs.

Whole server migration occurs when a server instance migrates to a different physical system upon failure.

Service-level migration is when services move to a different server instance within the cluster.

See these topics for server and service failover for JMS and JTA.

- About Automatic Service Migration (JMS/JTA)
  Service-level migration in WebLogic Server is the process of moving pinned services from one server instance to a different available server instance in the cluster.

About Automatic Service Migration (JMS/JTA)

Service-level migration in WebLogic Server is the process of moving pinned services from one server instance to a different available server instance in the cluster.

You can configure JMS and JTA services for high availability by using migratable targets. A **migratable target** is a special target that can migrate from one server in a cluster to another. A migratable target provides a way to group migratable services that should move together. High availability is achieved by migrating a migratable target from one clustered server to another when a problem occurs on the original server. When the migratable target migrates, all services that the target hosts migrate.

From release 12*c* (12.2.1.3.0), you can use the **High Availability Options** screen in the Configuration Wizard to automate the service-level migration configuration for the Fusion Middleware components. This screen appears for the first time when you create a cluster that uses Automatic Service Migration, persistent stores, or both, and all subsequent clusters that are added to the domain by using the Configuration Wizard, automatically apply the selected HA options.

If a product doesn't support Automatic Service Migration, you can use whole service migration.

> **Note:**
>
> See Service Migration in *Oracle Fusion Middleware Administering Clusters for Oracle* for these topics:
>
> - Understanding the Service Migration Framework
> - Pre-Migration Requirements
> - Roadmap for Configuring Automatic Migration of JMS-related Services
> - Roadmap for Configuring Automatic Migration of the JTA Transaction Recovery Service

Roadmap for Setting Up a High Availability Topology

Oracle recommends completing install and configuration steps in a certain order to set up an example high availability topology.

Table 2-6 describes high level steps required to set up an example middleware topology with high availability.

Table 2-6 Roadmap for Setting Up a High Availability Topology

Task	Description	Documentation
1. Install Real Application Clusters	Install Real Application Clusters	Install Oracle Database 12c Software with Oracle RAC in *Oracle Real Application Clusters Administration and Deployment Guide*

Table 2-6 (Cont.) Roadmap for Setting Up a High Availability Topology

Task	Description	Documentation
2. Install and configure middleware components	Install and configure the application by following instructions in an application installation guide.	Roadmap for Installing and Configuring the Standard Installation Topology in *Oracle Fusion Middleware Installing and Configuring the Oracle Fusion Middleware Infrastructure* or the installation guide for your product
3. Install and configure Oracle HTTP Server	Install and configure Oracle HTTP Server in the same domain	Roadmap for Installing and Configuring Oracle HTTP Server in a WebLogic Server Domain in *Installing and Configuring Oracle HTTP Server*
4. Configure a load balancer	Configure a third- party load balancer that meets specific requirements, or Oracle HTTP Server/Oracle Traffic Director.	Server Load Balancing in a High Availability Environment.
5. Scale out the topology (machine scale out)Oracle Fusion MiddlewareOracle Fusion Middleware	Steps for scaling out a topology (machine scale-out) for all products that are part of a Fusion Middleware WebLogic Server domain.	Scaling Out a Topology (Machine Scale Out)
6. Configure high availability for the Administration Server	Configure high availability for the Administration Server	Administration Server High Availability

3

Whole Server Migration

When **whole server migration** occurs, a server instance migrates to a different physical machine upon failure. You must configure whole server migration for Managed Servers if your environment uses special (pinned) services such as JMS and JTA.

- About Whole Server Migration
  WebLogic Server has **migratable servers** to make JMS and the JTA transaction system highly available.
- Configuring Whole Server Migration for Managed Server Failover
  If you configure Managed Server failover and one server in a cluster fails, whole server migration restarts a Managed Server on another machine.

About Whole Server Migration

WebLogic Server has **migratable servers** to make JMS and the JTA transaction system highly available.

A cluster provides high availability and failover by duplicating an object or service on redundant Managed Servers in the cluster. However, some services, such as JMS servers and JTA transaction recovery service, are designed with the assumption that there is only *one* active instance of the service in a cluster at any given time. These types of services are known as **pinned services** because they remain active on only one server at a time.

Most services deploy homogeneously on all Managed Servers in a cluster. With this setup, they can failover transparently from one Managed Server to another. However, pinned services target *one* Managed Server in a cluster. For pinned services, WebLogic Server supports failure recovery with migration instead of failover.

Migratable servers provide for both automatic and manual migration at the server-level, rather than the service level.

When a migratable server is unavailable for any reason (for example, if it hangs, loses network connectivity, or its host system fails), migration is automatic. Upon failure, a migratable server automatically restarts on the same system if possible. If a migratable server can't restart on the system it failed on, it migrates to another system. Also, you can manually start migration of a server instance.

> **Note:**
>
> See Whole Server Migration in *Oracle Fusion Middleware Administering Clusters for Oracle WebLogic Server* to prepare for automatic whole server migration, configure automatic whole server migration, and server migration processes and communications.
>
> See Service Details in *Oracle Fusion Middleware Administering Clusters for Oracle WebLogic Server* for details on service migration mechanisms.
>
> See JMS and JTA High Availability for details on JMS and JTA services.

Configuring Whole Server Migration for Managed Server Failover

If you configure Managed Server failover and one server in a cluster fails, whole server migration restarts a Managed Server on another machine.

Your system must meet specific requirements before you configure automatic whole server migration. See Preparing for Automatic Whole Server Migration in *Administering Clusters for Oracle WLS*.

To configure whole server migration, follow steps in Configuring Whole Server Migration in *Administering Clusters for Oracle WebLogic Server*.

See these topics for more on configuring whole server migration:

- Using High Availability Storage for State Data
- Server Migration Processes and Communications

Part II
Creating a High Availability Environment

Part II describes recommendations and steps you need to take to create a high availability environment.

This part contains the following topics:

- Using Shared Storage
- Database Considerations
- Scaling Out a Topology (Machine Scale Out)
- Using Dynamic Clusters
  A **dynamic cluster** is a cluster that contains one or more dynamic servers. A **dynamic server** is a server instance that gets its configuration from a server template. This is in contrast to Managed Servers, which require you to configure each server individually.
- JMS and JTA High Availability
- Administration Server High Availability

4
Using Shared Storage

Oracle recommends locating specific artifacts in shared storage for a high availability environment.
There are benefits to placing artifacts in a common location that multiple hosts or servers share. This common location typically resides in a shared file system, which is mounted on each server with standard operating system protocols such as NFS and CIFS.

- About Shared Storage
  Shared storage allows sharing of dynamic state and server configuration. It simplifies administration, configuration, failover, and backup/recovery.
- Shared Storage Prerequisites
  There are shared storage prerequisites that apply only when you use file-based persistent stores.
- Using Shared Storage for Binary (Oracle Home) Directories
  Oracle has guidelines for using shared storage for your Oracle home directories.
- Using Shared Storage for Domain Configuration Files
  There are guidelines for using shared storage for the Oracle WebLogic Server domain configuration files that you create when you configure Oracle Fusion Middleware products in an enterprise deployment.
- Shared Storage Requirements for JMS Stores and JTA Logs
  When you use file-based persistence in a high availability setup, you must configure the JMS persistent stores and JTA transaction log directories to reside in shared storage.
- Directory Structure and Configurations
  When you use shared storage, there are multiple ways to lay out storage elements. Oracle recommends certain best practices.

About Shared Storage

Shared storage allows sharing of dynamic state and server configuration. It simplifies administration, configuration, failover, and backup/recovery.

In a highly available environment, shared storage is *required* when you use file based persistent stores (for JMS and JTA logs) and certain Oracle products. Shared storage is *optional* for product binaries and domain directories.

The following artifacts are typical candidates to place on a shared file system:

- **Product binaries**: All files and directories related to product executables, JAR files, and scripts that install during product installation.
- **Domain directory**: Directory containing WebLogic Server domains and their configuration.
- **File-based persistent stores**: File-based persistent stores for JMS persistence and JTA transaction logs.

Table 4-1 has more information about shared storage.

Table 4-1 Shared Storage Topics

Topic/Task	For More Information
Structure and contents of an Oracle home	Understanding the Oracle Fusion Middleware Infrastructure Directory Structure in *Oracle Fusion Middleware Installing and Configuring the Oracle Fusion Middleware Infrastructure*
Saving JMS and JTA information in a file store	The WebLogic Persistent Store in *Administering the WebLogic Server Persistent Store*. Persistent Store High Availability in *Administering JMS Resources for Oracle WebLogic Server*. Default File Store Availability for JTA in *Administering Clusters for Oracle WebLogic Server*.

Shared Storage Prerequisites

There are shared storage prerequisites that apply only when you use file-based persistent stores.

Following are the list of shared storage prerequisites:

- For proper recovery in the event of a failure, you must store both JMS and JTA transaction logs in a location that is accessible to all nodes that can resume operations after a Managed Server failure. This setup requires a shared storage location that multiple nodes can reference. See Directory Structure and Configurations for the recommended directory structure.
- Oracle recommends that you use a shared storage device that is network-attached storage (NAS) or storage area network (SAN).

  If you use NFS-mounted systems, issues related to file locking and abrupt node failures have been detected. See Using File Stores on NFS and check with your storage vendor for the main recommended parameters for mount options.

  The following example command is based on a NAS device. Note: your options may be different from those in this example; see UNIX/Linux documentation for more on the mount command.

  ```
  mount nasfiler:/vol/vol1/u01/oracle /u01/oracle -t nfs -o
  rw,bg,hard,nointr,tcp,vers=3,timeo=300,rsize=32768,wsize=32768
  ```

- For maximum availability, Oracle recommends a *highly available* NAS or SAN device for shared storage. Shared storage devices that are not highly available can be a single point of failure. Check with your storage provider for options to achieve this.

For more on saving JMS and JTA information in a file store, see The WebLogic Persistent Store in *Administering the WebLogic Server Persistent Store*.

Using Shared Storage for Binary (Oracle Home) Directories

Oracle has guidelines for using shared storage for your Oracle home directories.

- About the Binary (Oracle Home) Directories
  When you install any Oracle Fusion Middleware product, you install product binaries into an Oracle home (*ORACLE_HOME*). The binary files are read-only

and don't change unless you patch or upgrade the Oracle home to a newer version

- About Sharing a Single Oracle Home
  You can configure multiple servers from one Oracle home. The benefit is that you can install the Oracle home in one location on a shared volume and reuse the Oracle home for multiple servers.
- About Using Redundant Binary (Oracle Home) Directories
  For maximum availability, Oracle recommends using redundant binary installations on shared storage.

About the Binary (Oracle Home) Directories

When you install any Oracle Fusion Middleware product, you install product binaries into an Oracle home (*ORACLE_HOME*). The binary files are read-only and don't change unless you patch or upgrade the Oracle home to a newer version

In a typical production environment, you save Oracle home files in a separate location from domain configuration files, which you create using the Configuration Wizard.

The Oracle home for an Oracle Fusion Middleware installation contains binaries for Oracle WebLogic Server, Oracle Fusion Middleware infrastructure files, and any Oracle Fusion Middleware product-specific directories.

> **Note:**
>
> The Configuration Wizard writes its logs to the `logs` directory in Oracle home. If you use a read-only Oracle home, you must specify the `-log` option to redirect logs to a different directory.

> **Note:**
>
> For more on the Oracle home structure and contents, see What are the Key Oracle Fusion Middleware Directories? in *Oracle Fusion Middleware Understanding Oracle Fusion Middleware Concepts*.

About Sharing a Single Oracle Home

You can configure multiple servers from one Oracle home. The benefit is that you can install the Oracle home in one location on a shared volume and reuse the Oracle home for multiple servers.

If multiple servers on different hosts share an Oracle home, there are some best practices to keep in mind. For example, because the Oracle inventory directory (`oraInventory`) is updated only on the host from which the Oracle home was originally installed, Oracle recommends that you perform all subsequent operations on the Oracle home (such as patching and upgrade) from that original host. If that host is unavailable, ensure that the Oracle inventory is updated on another host before you apply patches or upgrades to the Oracle home from the other host.

For more about `oraInventory`, see Oracle Universal Installer Inventory.

About Using Redundant Binary (Oracle Home) Directories

For maximum availability, Oracle recommends using redundant binary installations on shared storage.

In this model:

1. You install two identical Oracle homes for your Oracle Fusion Middleware software on two different shared volumes.
2. You then mount one of the Oracle homes to one set of servers and the other Oracle home to the remaining servers.

   Each Oracle home has the same mount point, so the Oracle home always has the same path, regardless of which Oracle home the server is using.

If one Oracle home becomes corrupted or unavailable, only half your servers are affected. For additional protection, Oracle recommends that you disk mirror these volumes. To restore affected servers to full functionality, you can simply remount the surviving Oracle Home.

If separate volumes are not available on shared storage, Oracle recommends simulating separate volumes using different directories within the same volume and mounting these to the same mount location on the host side. Although this doesn't guarantee the protection that multiple volumes provide, it does protect from user deletions and individual file corruption.

> **Note:**
>
> For maximum protection, Oracle recommends that you evenly distribute the members of a cluster across redundant binary Oracle homes. This is particularly important if cluster members are not running on all available servers.

Using Shared Storage for Domain Configuration Files

There are guidelines for using shared storage for the Oracle WebLogic Server domain configuration files that you create when you configure Oracle Fusion Middleware products in an enterprise deployment.

- About Oracle WebLogic Server Administration and Managed Server Domain Configuration Files
  When you configure an Oracle Fusion Middleware product, you create or extend an Oracle WebLogic Server domain. Each domain consists of a single Administration Server and one or more Managed Servers.
- Shared Storage Considerations for Administration Server Configuration Directory
  Oracle does not require you to store domain configuration files in shared storage. However, to support Administration Server recovery, you must place the Administration Server configuration directory on shared storage and mount it on the host that the Administration Server runs on.

- Shared Storage Considerations for Managed Server Configuration Files
  Oracle recommends that you keep Managed Server configuration files in local, or, host private, storage.

About Oracle WebLogic Server Administration and Managed Server Domain Configuration Files

When you configure an Oracle Fusion Middleware product, you create or extend an Oracle WebLogic Server domain. Each domain consists of a single Administration Server and one or more Managed Servers.

WebLogic uses a replication protocol to push persisted changes on the Administration Server to all Managed Servers. This gives redundancy to the Managed Servers so that you can start them without the Administration Server running. This mode is called *Managed Server independence*.

For more information about Oracle WebLogic Server domains, see Understanding Oracle WebLogic Server Domains.

Shared Storage Considerations for Administration Server Configuration Directory

Oracle does not require you to store domain configuration files in shared storage. However, to support Administration Server recovery, you must place the Administration Server configuration directory on shared storage and mount it on the host that the Administration Server runs on.

If that host fails, you can mount the directory on a different host and bring up the failed Administration Server on the other host. See Administration Server High Availability .

Shared Storage Considerations for Managed Server Configuration Files

Oracle recommends that you keep Managed Server configuration files in local, or, host private, storage.

You can keep Managed Server configuration files on shared storage. However, doing so can affect performance because multiple servers concurrently access the same storage volume.

Shared Storage Requirements for JMS Stores and JTA Logs

When you use file-based persistence in a high availability setup, you must configure the JMS persistent stores and JTA transaction log directories to reside in shared storage.

See Using File Persistence.

Directory Structure and Configurations

When you use shared storage, there are multiple ways to lay out storage elements. Oracle recommends certain best practices.

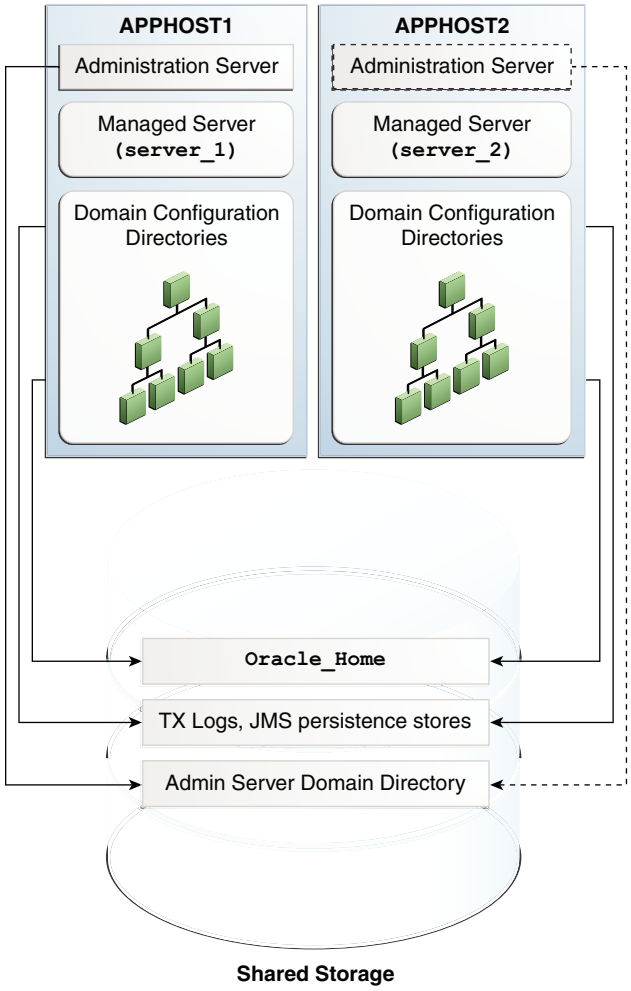
Table 4-2 Shared Storage Elements Directory Structure

Element	Location
ORACLE_HOME	Share in read-only mode by all servers.
JMS file stores and Transaction logs	Place on shared storage if you use file-based persistence.
Administration Server domain configuration directory	Place in shared storage to facilitate failing over the Administration Server to a different host.

> **Note:**
>
> Place Managed Server domain configuration directories on storage that is local to the corresponding host. See Shared Storage Considerations for Administration Server Configuration Directory.

Figure 4-1 illustrates the directory structure.

Figure 4-1 Shared Storage Directory Structure



5

Database Considerations

As you configure database connections for Oracle Fusion Middleware in a high availability setup, you must make decisions about Oracle Real Application Clusters (Oracle RAC). Oracle RAC is a commonly-deployed database high availability solution. Most components use a database as the persistent store for their data. When you use an Oracle database, you can configure it in a variety of highly available configurations.

For more information about database options, see Oracle Database High Availability Overview in *High Availability Overview and Best Practices*.

- About Oracle Real Application Clusters
  A **cluster** comprises multiple interconnected computers or servers that appear as one server to end users and applications. With Oracle RAC, you can cluster an Oracle database so that it is highly scalable and highly available.
- Roadmap for Setting Up Oracle Real Application Clusters
  Use this roadmap to set up Oracle RAC.
- About RAC Database Connections and Failover
  To establish connection pools, Oracle Fusion Middleware supports Active GridLink data sources and multi data sources for the Oracle RAC back end (for both XA and non-XA JDBC drivers). These data sources also support load balancing across Oracle RAC nodes.
- About Data Sources
  A **data source** is an abstraction that components use to obtain connections to a relational database.
- Configuring Active GridLink Data Sources with Oracle RAC
  You configure component data sources as Active GridLink data sources for a RAC database during domain creation.
- Configuring Multi Data Sources
  There are multiple tools available to configure multi data sources.

About Oracle Real Application Clusters

A **cluster** comprises multiple interconnected computers or servers that appear as one server to end users and applications. With Oracle RAC, you can cluster an Oracle database so that it is highly scalable and highly available.

All Oracle Fusion Middleware components that you deploy to Oracle WebLogic Server support Oracle RAC.

Every Oracle RAC instance in a cluster has equal access and authority. Node and instance failure may affect performance, but doesn't result in downtime; the database service is available or can be made available on surviving server instances.

Roadmap for Setting Up Oracle Real Application Clusters

Use this roadmap to set up Oracle RAC.

Table 5-1 outlines tasks and information to set up Oracle RAC.

Table 5-1 Roadmap for Setting up Oracle RAC

Task/Topic	More Information
About Oracle RAC	Introduction to Oracle RAC in *Oracle Real Application Clusters Administration and Deployment Guide*
Installing Oracle RAC	*Oracle Real Application Clusters Administration and Deployment Guide*
Managing Oracle RAC	Overview of Managing Oracle RAC Environments in *Oracle Real Application Clusters Administration and Deployment Guide*
Configuring and tuning GridLink and multi data sources	*Oracle Fusion Middleware Configuring and Managing JDBC Data Sources for Oracle WebLogic Server*
Configuring Single Client Access Name (SCAN) URLs. (To specify the host and port for TNS and ONS listeners in WebLogic console.)	SCAN Addresses in *Oracle Fusion Middleware Configuring and Managing JDBC Data Sources for Oracle WebLogic Server*.

About RAC Database Connections and Failover

To establish connection pools, Oracle Fusion Middleware supports Active GridLink data sources and multi data sources for the Oracle RAC back end (for both XA and non-XA JDBC drivers). These data sources also support load balancing across Oracle RAC nodes.

If an Oracle RAC node or instance fails, WebLogic Server or Oracle Thin JDBC driver redirects session requests to another node in the cluster. Existing connections don't failover. However, new application connection requests are managed using existing connections in the WebLogic pool or by new connections to the working Oracle RAC instance.

If the database is the transaction manager, in-flight transactions typically roll back.

If WebLogic Server is the transaction manager, in-flight transactions fail over; they are driven to completion or rolled back based on the transaction state when failure occurs.

For more on XA Transactions, see About XA Transactions.

- About XA Transactions
  XA transaction support enables access to multiple resources (such as databases, application servers, message queues, transactional caches) within one transaction.

About XA Transactions

XA transaction support enables access to multiple resources (such as databases, application servers, message queues, transactional caches) within one transaction.

A *non-XA transaction* always involves just one resource.

An XA transaction involves a coordinating transaction manager with one or more databases, or other resources such as JMS, all involved in a single global transaction.

Java EE uses the terms **JTA transaction**, **XA transaction**, **user transaction**, and **global transaction** interchangeably to refer to one global transaction. This transaction type may include operations on multiple different XA- capable or non-XA resources and even different resource types. A JTA transaction is always associated with the current thread, and may pass from server to server as one application calls another. A common example of an XA transaction is one that includes both a WebLogic JMS operation and a JDBC (database) operation.

About Data Sources

A **data source** is an abstraction that components use to obtain connections to a relational database.

Connection information, such as the URL or user name and password, is set on a data source object as properties. The application's code does not need to explicitly define the properties. Due to this abstraction, you can build applications in a portable manner, because they are not tied to a specific back-end database. The database can change without affecting application code.

Active GridLink data sources and multi data sources support database connection high availability, load balancing, and failover. Oracle recommends the following data source types depending on your Oracle RAC database version:

- If you use Oracle RAC database version 11g Release 2 and later, use Active GridLink data sources.
- If you use an Oracle RAC database version earlier than 11g Release 2 or a non-Oracle database, use multi data sources.

> **Note:**
>
> Oracle recommends using Active GridLink data sources with Oracle RAC database for maximum availability. For Oracle RAC database versions that don't support Active GridLink data sources, Oracle recommends using multi data sources for high availability.

- Active GridLink Data Sources
  An **Active GridLink data source** provides connectivity between WebLogic Server and an Oracle database service, which may include multiple Oracle RAC clusters.
- Multi Data Sources
  A **multi data source** is an abstraction around a group of data sources that provides load balancing or failover processing. Multi data sources support load balancing for both XA and non-XA data sources.

Active GridLink Data Sources

An **Active GridLink data source** provides connectivity between WebLogic Server and an Oracle database service, which may include multiple Oracle RAC clusters.

An Active GridLink data source has features of generic data sources, plus the following support for Oracle RAC:

- Uses the ONS to respond to state changes in an Oracle RAC.
- Responds to Fast Application Notification (FAN) events to provide Fast Connection Failover (FCF), Runtime Connection Load-Balancing, and RAC instance graceful shutdown. FAN is a notification mechanism that Oracle RAC uses to quickly alert applications about configuration and workload.
- Provides Affinities (or XA Affinity) policies to ensure all database operations for a session are directed to the same instance of a RAC cluster for optimal performance.
- SCAN Addresses
- Secure Communication Using Oracle Wallet

See Using Active GridLink Data Sources in *Oracle Fusion Middleware Configuring and Managing JDBC Data Sources for Oracle WebLogic Server* for more on the following topics:

- What is an Active GridLink Data Source
- Using Socket Direct Protocol
- Configuring Connection Pool Features
- Configuring Oracle Parameters
- Configuring an ONS Client
- Tuning Active GridLink Data Source Connection Pools
- Monitoring GridLink JDBC Resources

Multi Data Sources

A **multi data source** is an abstraction around a group of data sources that provides load balancing or failover processing. Multi data sources support load balancing for both XA and non-XA data sources.

A **failover multi data source** provides an ordered list of data sources to fulfill connection requests. Normally, every connection request to this kind of multi data source is served by the first data source in the list.

A **load-balancing multi data source** chooses from a circular list of datasources in a round robin method. The stream of incoming connection requests is spread evenly around the datasources. If a database connection test fails and the connection can't be replaced, or if the data source is suspended, a connection is sought from the next data source on the list.

Multi data sources are bound to the JNDI tree or local application context just like regular data sources. Applications look up a multi data source on the JNDI tree or in the local application context (`java:comp/env`) just as they do for data sources, and then request a database connection. The multi data source determines which data source to use to satisfy the request depending on the algorithm selected in the multi data source configuration: load balancing or failover.

> **Note:**
>
> To configure Multi Data Sources with Oracle RAC, see Using Multi Data Sources with Oracle RAC in *Oracle Fusion Middleware Configuring and Managing JDBC Data Sources for Oracle WebLogic Server*.

Configuring Active GridLink Data Sources with Oracle RAC

You configure component data sources as Active GridLink data sources for a RAC database during domain creation.

How you configure an Active GridLink data source depends on:

- The Oracle component that you are working with
- The domain you are creating

> **Note:**
>
> To create and configure Active GridLink data sources, see Using Active GridLink Data Sources in *Oracle Fusion Middleware Configuring and Managing JDBC Data Sources for Oracle WebLogic Server*.

- Requirements to Configure Component Data Sources as Active Gridlink Data Sources
  Your system must meet certain requirements before you configure component data sources as Active GridLink data sources to use with an Oracle RAC database.
- Configuring Component Data Sources as Active GridLink Data Sources
  You configure component data sources as Active GridLink data sources for a RAC database during domain creation.
- Using SCAN Addresses for Hosts and Ports
  Oracle recommends using Oracle Single Client Access Name (SCAN) addresses to specify the host and port for the TNS listener and ONS listener in the WebLogic console.

Requirements to Configure Component Data Sources as Active Gridlink Data Sources

Your system must meet certain requirements before you configure component data sources as Active GridLink data sources to use with an Oracle RAC database.

- You use Oracle RAC database version 11g Release 2 or later.
- You have run RCU to create component schemas.
- You use the Configuration Wizard to create or configure a domain and have arrived at the JDBC Component Schema screen where you select **Component Datasources**.

Configuring Component Data Sources as Active GridLink Data Sources

You configure component data sources as Active GridLink data sources for a RAC database during domain creation.

To configure component data sources as Active GridLink data sources:

1. In the JDBC Component Schema screen, select one or more component schemas to configure GridLink data sources for.
2. Select **Convert to GridLink** then select **Next**.
3. In the GridLink Oracle RAC Component Schema screen, select one of the GridLink JDBC drivers.
4. In the **Service Name** field, enter the service name of the database using lowercase characters. For example, `mydb.example.com`.
5. In the **Schema Owner** field, enter the name of the database schema owner for the corresponding component.
6. In the **Schema Password** field, enter the password for the database schema owner.
7. In the **Service Listener**, **Port**, and **Protocol** field, enter the SCAN address and port for the RAC database being used. The protocol for Ethernet is TCP; for Infiniband it is SDP. Click **Add** to enter multiple listener addresses.

   You can identify the SCAN address by querying the appropriate parameter in the database using the TCP protocol:

   ```
   show parameter remote_listener


   NAME TYPE VALUE

   ------------------------------------------------------------------------

   remote_listener string db-scan.example.com:1521
   ```

   You can also identify the SCAN address by using the `srvctl config scan` command. Use the command `srvctl config scan_listener` to identify the SCAN listener port.
8. Select **Enable FAN** to receive and process FAN events. Enter one or more ONS daemon listen addresses and port information. Select **Add** to enter more entries.

> **Note:**
>
> Verify that the ONS daemon listen address(es) that you enter is valid. The domain creation process do not validate the address.

To determine the Scan (ONS) port, use the RAC `srvctl` command on the Oracle Database server, as the following example shows:

```
srvctl config nodeapps -s

ONS exists: Local port 6100, remote port 6200, EM port 2016
```

9. Select **Enable SSL** for SSL communication with ONS. Enter the Wallet File, which has SSL certificates, and the Wallet Password.
10. Select **Next**. Verify that all connections are successful.

> **Note:**
>
> See:
>
> - JDBC Component Schema in *Oracle Fusion Middleware Creating WebLogic Domains Using the Configuration Wizard* for information about the JDBC Component Schema screen.
> - GridLink Oracle RAC Component Schema in *Oracle Fusion Middleware Creating WebLogic Domains Using the Configuration Wizard* for information about configuring component schemas.
> - Using Active GridLink Data Sources in *Administering JDBC Data Sources for Oracle WebLogic Server* for information on GridLink RAC data sources.

Using SCAN Addresses for Hosts and Ports

Oracle recommends using Oracle Single Client Access Name (SCAN) addresses to specify the host and port for the TNS listener and ONS listener in the WebLogic console.

You do not need to update an Active GridLink data source containing SCAN addresses if you add or remove Oracle RAC nodes. Contact your network administrator for appropriately configured SCAN URLs for your environment. See SCAN Addresses in *Oracle Fusion Middleware Configuring and Managing JDBC Data Sources for Oracle WebLogic Server*.

Configuring Multi Data Sources

There are multiple tools available to configure multi data sources.

- Oracle Fusion Middleware Configuration Wizard (during WebLogic Server domain creation)
- Oracle WebLogic Server Administration Console
- WLST Commands
- Configuring Multi Data Sources with Oracle RAC
  Configuring Multi Data Sources with Oracle RAC has specific requirements and steps that you must complete.
- Configuring Multi Data Sources for MDS Repositories
  You can configure applications that use an MDS database-based repository for high availability Oracle database access.

Configuring Multi Data Sources with Oracle RAC

Configuring Multi Data Sources with Oracle RAC has specific requirements and steps that you must complete.

- Requirements to Configure Multi Data Sources with Oracle RAC
  Verify that your system meets the following requirements before you configure component data sources as multi data sources to use with an Oracle RAC database.
- Configuring Component Data Sources as Multi Data Sources
  When you configure component data sources as multi data sources, you select data sources to convert then enter database information.
- About Adding Multi Data Sources For RAC Databases
  Multi data sources have constituent data sources for each RAC instance that provides a database service. If you add an instance to the RAC back end, Oracle recommends adding an additional data source to the multi data source on the Fusion Middleware tier.
- Modifying or Creating Multi Data Sources After Initial Configuration
  For multi data sources that you create manually or modify after initial configuration, Oracle strongly recommends specific XA and non-XA data source property values for optimal high availability. Make changes only after careful consideration and testing if your environment requires that you do so.
- Troubleshooting Warning Messages (Increasing Transaction Timeout for XA Data Sources)
  If WARNING messages in server logs have the following exception, you may need to increase the XA timeout value in your setup.
- Configuring Schemas for Transactional Recovery Privileges
  You want to enable WebLogic Server transaction manager to perform schema tasks.

Requirements to Configure Multi Data Sources with Oracle RAC

Verify that your system meets the following requirements before you configure component data sources as multi data sources to use with an Oracle RAC database.

- You are using an Oracle RAC database.
- You have run RCU to create component schemas.
- You are using the Configuration Wizard to create or configure a domain and have arrived at the JDBC Component Schema Screen where you select Component Schemas. Before you arrive at the JDBC Component Schema screen, you must select the option **Manual Configuration** in the Database Configuration Type screen.

Configuring Component Data Sources as Multi Data Sources

When you configure component data sources as multi data sources, you select data sources to convert then enter database information.

To configure component data sources as multi data sources:

1. In the Component Datasources screen, select one or more component schemas to configure RAC Multiple data sources for.
2. Select **Convert to RAC multi data source** then select **Next**.
3. In the Oracle RAC Multi Data Source Component Schema screen, the JDBC driver **Oracle's Driver (Thin) for RAC Service-Instance connections; Versions:10 and later**.
4. In the **Service Name** field, enter the database service name enter in lowercase, for example, `mydb.example.com`.
5. In the **Schema Owner** field, enter the username of the database schema owner for the corresponding component.
6. In the **Schema Password** field, enter the password for the database schema owner.
7. In the **Host Name**, **Instance Name**, and **Port** field, enter the RAC node hostname, database instance name, and port. Click **Add** to enter multiple listener addresses.
8. Click **Next**.Verify that all connections are successful.

About Adding Multi Data Sources For RAC Databases

Multi data sources have constituent data sources for each RAC instance that provides a database service. If you add an instance to the RAC back end, Oracle recommends adding an additional data source to the multi data source on the Fusion Middleware tier.

When you migrate a database from a non-RAC to a RAC database, you must create an equivalent, new multi data source for each affected data source. Multi data sources that you create must have constituent data sources for each RAC instance. Data source property values must be identical to the original single instance data source for properties in Configuring Multi Data Sources with Oracle RAC. For example, if a single instance data source driver is `oracle.jdbc.xa.client.OracleXADataSource`, it must be `oracle.jdbc.xa.client.OracleXADataSource` for each constituent data source of the new multi data source.

Modifying or Creating Multi Data Sources After Initial Configuration

For multi data sources that you create manually or modify after initial configuration, Oracle strongly recommends specific XA and non-XA data source property values for optimal high availability. Make changes only after careful consideration and testing if your environment requires that you do so.

Table 5-2 describes XA and non-XA data source property values that Oracle recommends.

Table 5-2 Recommended Multi Data Source Configuration

Property Name	Recommended Value
test-frequency-seconds	5
algorithm-type	Load-Balancing

For individual data sources, Oracle recommends the configuration values in Table 5-3 for high availability environments. Oracle recommends that you set any other parameters according to application requirements.

Table 5-3 XA Data Source Configuration

Property Name	Recommended Value
Driver	oracle.jdbc.xa.client.OracleXADataSource
Property command	<property> <name>oracle.net.CONNECT_TIMEOUT</name> <value>10000</value> </property>
initial-capacity	0
connection-creation-retry-frequency-seconds	10
test-frequency-seconds	300
test-connections-on-reserve	true
test-table-name	SQL SELECT 1 FROM DUAL
seconds-to-trust-an-idle-pool-connection	0
global-transactions-protocol	TwoPhaseCommit
keep-xa-conn-till-tx-complete	true
xa-retry-duration-seconds	300
xa-retry-interval-seconds	60

Troubleshooting Warning Messages (Increasing Transaction Timeout for XA Data Sources)

If WARNING messages in server logs have the following exception, you may need to increase the XA timeout value in your setup.

```
[javax.transaction.SystemException: Timeout during commit processing
```

To increase the transaction timeout for the XA Data Sources setting, use the Administration Console:

1. Access the data source configuration.
2. Select the **Transaction** tab.
3. Set the XA Transaction Timeout to a larger value, for example, **300**.
4. Select the **Set XA Transaction Timeout** checkbox. You *must* select this checkbox for the new XA transaction timeout value to take effect.
5. Click **Save**.

Repeat this configuration for all individual data sources of an XA multi data source.

Table 5-4 Non-XA Data Source Configuration

Property Name	Recommended Value
Driver	oracle.jdbc.OracleDriver

Table 5-4 (Cont.) Non-XA Data Source Configuration

Property Name	Recommended Value
Property to set	<property> <name>oracle.net.CONNECT_TIMEOUT</name> <value>10000</value> </property>
initial-capacity	0
connection-creation-retry-frequency-seconds	10
test-frequency-seconds	300
test-connections-on-reserve	true
test-table-name	SQL SELECT 1 FROM DUAL
seconds-to-trust-an-idle-pool-connection	0
global-transactions-protocol	None

Configuring Schemas for Transactional Recovery Privileges

You want to enable WebLogic Server transaction manager to perform schema tasks.

You must have sysdba privileges to enable transaction manager privileges:

- Query for transaction state information.
- Issue the appropriate commands, such as commit and rollback, during recovery of in-flight transactions after a WebLogic Server container failure.

To configure schemas for transactional recovery privileges:

1. Log on to SQL*Plus as a user with sysdba privileges. For example:

   ```
   sqlplus "/ as sysdba"
   ```

2. Grant select on `sys.dba_pending_transactions` to the `appropriate_user`.
3. Grant force any transaction to the `appropriate_user`.

Configuring Multi Data Sources for MDS Repositories

You can configure applications that use an MDS database-based repository for high availability Oracle database access.

With this configuration, failure detection, recovery, and retry by MDS (and by WebLogic infrastructure) protect application read-only MDS operations from Oracle RAC database planned and unplanned downtimes

The Fusion Middleware Control navigation tree exposes multi data sources as MDS repositories. You can select these multi data sources when you customize application deployment and use them with MDS WLST commands.

- Configuring an application to retry read-only operations

  To configure an application to retry the connection, configure the `RetryConnection` attribute of the application's MDS AppConfig MBean. See *Oracle Fusion Middleware Administrator's Guide*.

- Registering an MDS multi data source

  In addition to steps in Configuring Multi Data Sources with Oracle RAC, note the following:

  – You must configure child data sources that comprise a multi data source used for an MDS repository as non-XA data sources.

  – A multi data source's name must have the prefix `mds-`. This ensures it is recognized as an MDS repository.

  > **Note:**
  >
  > When you add a MDS data source as a child of a multi data source, this data source is no longer exposed as an MDS repository. It does not appear under the Metadata Repositories folder in the Fusion Middleware Control navigation tree. You can not perform MDS repository operations on it and it does not appear in the list of selectable repositories during deployment.

- Converting a data source to a multi data source

  Keep in mind when you convert a data source to a multi data source:

  – To create a new multi data source with a new, unique name, redeploy the application and select this new multi data source as the MDS repository during deployment plan customization.

  – To avoid redeploying the application, you can delete the data source and recreate a new multi data source using the same name and jndi-name attributes.

6

Scaling Out a Topology (Machine Scale Out)

Steps to scale out a topology (machine scale-out) are similar for all Fusion Middleware products that are a part of a WebLogic Server domain. To enable high availability, it is important to provide failover capabilities to another host computer. When you do so, your environment can continue to serve your application consumers if a computer goes down.

- About Machine Scale Out
  Scalability is the ability of a piece of hardware or software, or a network, to *expand* or *shrink* to meet future needs and circumstances. A scalable system can handle increasing numbers of requests without adversely affecting response time and throughput.
- Roadmap for Scaling Out Your Topology
  When your product is installed, configured, and has a cluster of Managed Servers, use this roadmap to scale out your topology.
- Optional Scale Out Procedure
  If you follow a standard installation topology (SIT), you have multiple Managed Servers assigned to a single host computer.
- About Scale Out Prerequisites
  Before you start the scale out process, you must have a **standard installation topology** (SIT) set up for your product. A SIT is the starting point for scale out.
- Resource Requirements
  Before you scale out the topology, verify that your environment meets certain requirements.
- Creating a New Machine
  A **machine** is the logical representation of the computer that hosts one or more WebLogic Server instances (Managed Servers). In a WebLogic domain, the machine definitions identify physical units of hardware and are associated with Managed Servers that they host.
- Configuring WLS JMS After Machine Scale Up or Scale Out
  You must manually recreate all JMS system resources on a new Managed Server when you add one to a cluster.
- Packing the Domain on APPHOST1
  You create a Managed Server template by running the `pack` command on a WebLogic domain.
- Preparing the New Machine
  To prepare the new machine, **machine_2**, verify that **APPHOST2** can access the shared disk where the Oracle home is installed, or install the same software that you installed on **machine_1**.
- Running Unpack to Transfer the Template
  To unpack the template and transfer the *domain_name*.jar file from **APPHOST1** to **APPHOST2**, run the unpack command.
- Starting the Node Manager
  You use Node Manager to start Managed Servers (using the Administration Console or Fusion Middleware Control).

- Starting Managed Servers
  You start Managed Servers in the Administration Console.
- Verifying Machine Scale Out
  To determine if the machine scale out succeeded, verify that the server status is **RUNNING** (after you use Administration Console to start it).
- Configuring Multicast Messaging for Clusters
  You configure clusters to use messaging so that they can communicate whether or not services are available and other information.

About Machine Scale Out

Scalability is the ability of a piece of hardware or software, or a network, to *expand* or *shrink* to meet future needs and circumstances. A scalable system can handle increasing numbers of requests without adversely affecting response time and throughput.

Machine scale-out is moving a server, one of many on one machine, to another machine for high availability. Machine scale out is different from Managed Server **scale up**, which is adding a new Managed Server to a machine that already has one or more Managed Servers running on it. See Scaling Up Your Environment in *Oracle Fusion Middleware Administering Oracle Fusion Middleware*.

Roadmap for Scaling Out Your Topology

When your product is installed, configured, and has a cluster of Managed Servers, use this roadmap to scale out your topology.

Table 6-1 describes typical steps you must take to scale out a topology.

If you already have a SIT (as *Installing and Configuring the Oracle Fusion Middleware Infrastructure* and product installation guides such as *Installing and Configuring Oracle SOA Suite and Business Process Management* describe), you do *not* need to repeat Resource Requirements, Creating a New Machine, and Configuring WLS JMS After Machine Scale Up or Scale Out steps.

Table 6-1 has start-to-finish steps if you did not complete SIT steps.

Table 6-1 Roadmap for Scaling Out Your Topology

Task	Description	More Information
Product is ready for scale out	Product is installed, configured, and a cluster of Managed Servers is available; your product is in a SIT	About Scale Out Prerequisites
Verify that you meet resource requirements	You must verify that your environment meets certain requirements	Resource Requirements
Create a new machine and assign servers to it	Use the Administration Console to create a new machine and add Managed Servers to it.	Creating a New Machine

Table 6-1 (Cont.) Roadmap for Scaling Out Your Topology

Task	Description	More Information
Create a new JMS server and target it	Create a new JMS server and target it to the new Managed Server	Configuring WLS JMS After Machine Scale Up or Scale Out
Run the pack command	Pack up the domain directory	Packing the Domain on APPHOST1
Prepare the new machine	Install the same software that you installed on the first machine	Preparing the New Machine
Run the unpack command	Create a Managed Server template.	Running Unpack to Transfer the Template
Start the server	Starts the Managed Server on the new machine	Starting Managed Servers
Verify the topology	Test the new setup	Verifying Machine Scale Out
Set the cluster messaging mode to Multicast	Modifies messaging mode from Unicast to Multicast (preferred for multi-server domains)	Configuring Multicast Messaging for Clusters

> **Note:**
>
> *APPHOST* refers to a physical host computer. *Machine* refers to the WebLogic Server machine definition describing that host. See Understanding the Oracle Fusion Middleware Infrastructure Standard Installation Topology in *Installing and Configuring the Oracle Fusion Middleware Infrastructure*.

Optional Scale Out Procedure

If you follow a standard installation topology (SIT), you have multiple Managed Servers assigned to a single host computer.

This set up is the most flexible way to create and scale out a domain topology so that it can meet changing requirements. It allows you to 1) create and validate a single-host domain, which is targeted to a single machine on a single host computer, and then 2) *retarget* the Managed Servers to additional machines, when additional computing resources are required. Also, it facilitates troubleshooting; you can validate the basic domain then scale up and scale out (and troubleshoot) at a later time.

However, if you know ahead of time what your target topology is, you can create additional machines during domain creation then just run pack and unpack steps.

If you assigned Managed Servers to target machines during installation or through an online administrative operation, skip Creating a New Machine when you go through the roadmap (Roadmap for Scaling Out Your Topology).

See Creating a New Machine in your product installation guide for more on machine mapping.

About Scale Out Prerequisites

Before you start the scale out process, you must have a **standard installation topology** (SIT) set up for your product. A SIT is the starting point for scale out.

If you followed the steps in your product installation guide, you should have a SIT. For an example, see the SIT that Understanding the Oracle Fusion Middleware Infrastructure Standard Installation Topology describes in *Oracle Fusion Middleware Installing and Configuring the Oracle Fusion Middleware Infrastructure*.

> **Note:**
>
> For more on the SIT, see your product's installation guide or About the Standard Installation Topology in *Planning an Installation of Oracle Fusion Middleware*.

> **Note:**
>
> Application tier products in this release don't support dynamic clusters. **Dynamic clusters** consist of server instances that can dynamically scale up to meet your application resource needs. A dynamic cluster uses a single server template to define configuration for a specified number of generated (dynamic) server instances.

Resource Requirements

Before you scale out the topology, verify that your environment meets certain requirements.

- At least one machine runs multiple Managed Servers configured with a product. This is the result of following your product installation guide or administration guide to add additional servers.
- A host computer in addition to your starting host computer.
- Each host computer can access the Oracle home that contains the product binaries by one of the following means:
  - Shared disk with binaries from the original installation
  - Dedicated disk with a new installation (matches the original installation)
  - Dedicated disk with a clone of the binaries from the original installation

  See Using Shared Storage.
- Sufficient storage available for the domain directory.
- Access to the same Oracle or third-party database used for the original installation.
- A shared disk for JMS and transaction log files (required when using a file persistence profile).

Creating a New Machine

A **machine** is the logical representation of the computer that hosts one or more WebLogic Server instances (Managed Servers). In a WebLogic domain, the machine definitions identify physical units of hardware and are associated with Managed Servers that they host.

Follow steps in this section to create a new machine.

- Shutting Down the Managed Server
  The Managed Server must be shut down before moving to a new machine.
- Creating a New Machine (Using the Administration Console)
  You create a new machine to host Managed Servers and identify physical units of hardware. Typically you use the Administration Console to create a new machine.
- Assigning Managed Servers to a New Machine
  You assign Managed Servers to the new machine to specify which servers the machine hosts.

Shutting Down the Managed Server

The Managed Server must be shut down before moving to a new machine.

If it is up and running, see the topic Shut Down a Server Instance in Administration Console Online Help to shut down the Managed Server that the new machine will host. See Shut Down Server Instances in a Cluster if the Managed Server is in a cluster.

Creating a New Machine (Using the Administration Console)

You create a new machine to host Managed Servers and identify physical units of hardware. Typically you use the Administration Console to create a new machine.

To use WLST to create a new machine, see Creating a New Machine for Certain Components in *Oracle Fusion Middleware Administering Oracle Fusion Middleware*.

> **Note:**
>
> The machine you create in this procedure must have a listen address on a specific network interface, not just a local host.

To create a new machine in the domain:

1. Start the domain Administration Server if it isn't running. Go to the *DOMAIN_HOME*/bin directory and run:

   ```
   ./startWeblogic.sh
   ```

2. When the Administration Server is running, access the Administration Console. Open a web browser and enter the URL:

   ```
   http://hostname:port/console
   ```

   On the Welcome screen, log in.

3. In the Administration Console Change Center, click **Lock & Edit**.

> **Note:**
>
> For production domains, Oracle recommends creating the domain in **Production** mode, which enables Change Center. If Production mode is *not* enabled, Change Center steps are not required.

4. Under Domain Structure, expand Environment then click **Machines**.
5. Above the Machines table (above Summary), click **New**.
6. In the Create a New Machine screen, enter a name for the machine, such as **machine_2**. Select the **Machine OS** using the drop-down list then click **Next**.
7. On the next screen, for **Type:**, use the drop-down list to select **Plain**.

   For the Node Manager **Listen Address**, enter the IP address or host name of the host computer that will represent this machine. Both machines appear in the Machines table.
8. Click **Finish**.
9. In the Change Center, click **Activate Changes**.

   The message **All changes have been activated. No restarts are necessary.** opens to indicate that you have a new machine.

Assigning Managed Servers to a New Machine

You assign Managed Servers to the new machine to specify which servers the machine hosts.

To add Managed Servers to a newly-created machine, use the Administration Console:

1. In the Change Center, click **Lock & Edit**.
2. In the Machines table, select the checkbox for **machine_1**.
3. Click the machine name (represented as a hyperlink).
4. Under the Settings for **machine_1**, click the Configuration tab then the Servers subtab.
5. Above the Servers table, click **Add**.
6. On the Add a Server to Machine screen, click **Create a new server and associate it with this machine**. Click **Next** then enter the **Server Name** and the **Server Listen Port** in the fields (required).
7. Under Domain Structure, click **Machines**.

   In the Machines table, click the machine **machine_2**.
8. Under the Settings for **machine_2**, click the Configuration tab and then the Servers tab. Above the Servers tab, click **Add**.

   On the **Add a Server to Machine** screen, select the button **Select an existing server**, **and associate it with this machine**.

   Use the Select a server drop-down list to choose **server_2** then select **Finish**.

   The message **Server created successfully** appears.

9. Verify that all Managed Servers have the correct Server Listen Address. Under Domain Structure, click **Servers**. In the Servers table, click the name of the Managed Server. Select the Configuration tab. Verify/set the Listen Address to the IP address of the machine it is associated with. Click **Save**.

10. To complete the changes, go back to the Change Center. Click **Activate Changes**. The message **All changes have been activated. No restarts are necessary.** appears.

    To see a summary of Managed Server assignments, under Domain Structure, under Environment, click **Servers**. The Servers table shows all servers in the domain and their machine assignments.

Configuring WLS JMS After Machine Scale Up or Scale Out

You must manually recreate all JMS system resources on a new Managed Server when you add one to a cluster.

New JMS system resources are cloned from an existing Managed Server in the cluster. New JMS system resources must have unique names in the domain. When you create a domain, the Configuration Wizard creates JMS system resource names that follow a pattern. For ease of use and manageability, Oracle recommends that you follow the same naming pattern.

To configure JMS resources on a new Managed Server `server_n`:

1. In the **Domain Structure** tree, select **Services** then select **Messaging**. Select **JMS Servers** to open the JMS Servers table.
2. In the Name column, identify all JMS servers that target one of the existing Managed Servers in the cluster, for example, `server_1`.

   The JMS server name format is *ResourceName*_**auto**_*number*.

   - *ResourceName* is the resource's identifying name
   - *number* identifies the JMS server; servers with the suffix 1 or 2 were created when the domain was created.
3. Note the name of the JMS server and its persistent store. For example, `UMSJMSServer_auto_1` and `UMSJMSFileStore_auto_1`.
4. Click **New** to create a new JMS server.
   a. Name the JMS server for `server_n` using the same format as `server_1`. For example, `UMSJMSServer_auto_n`.
   b. For the **Persistent Store**, click **Create a New Store**.
   c. For **Type**, select the same persistence profile used that the JMS Server uses on `server_1`. Click **Next**.
   d. Enter a persistent store in the **Name** field. Use the same format as the `server_1` persistent store. For example, `UMSJMSFileStore_auto_n`.
   e. In the **Target** field, select the migratable target for migratable target `server_n (migratable)` from the drop down list.
   f. In the **Directory** field, use the same name as the persistent store. Click **OK** to create the persistent store.
   g. In the Create a New JMS Server screen, select the new persistent store and click **Next**.

h. For JMS Server Target, select the migratable target for `server_n` (migratable) from the drop down list.

i. Click **Finish** to create the JMS server.

5. Update Subdeployment targets for the corresponding JMS Modules to include the new JMS server:

a. In the Administration Console's **Domain Structure** tree, expand the **Services** node, expand the **Messaging** node, and select **JMS Modules** to open the JMS Modules table.

b. Identify the JMS system module that corresponds to the JMS server; its name has a common root. For example for JMS server `UMSJMSServer_auto_1`, the JMS module name is `UMSJMSSystemResource`. Click on the JMS module (a hyperlink) in the Name column to open the Module Settings page.

c. Open the Subdeployments tab and click on the subdeployment for the JMS module in the Name column.

d. Select the new JMS Server `server_n`. Click **Save**.

e. Go back to the module Settings page. Select the **Targets** tab. Verify that **All servers in the cluster** is enabled.

f. Click **Save** if you changed anything.

You now have a new Managed Server that has configured JMS resources in the cluster.

Packing the Domain on APPHOST1

You create a Managed Server template by running the `pack` command on a WebLogic domain.

> **Note:**
>
> The Administration Server should be running on APPHOST1 when you go through pack and unpack steps.

Run the `pack` command on **APPHOST1** to create a template pack. You unpack the template file on **APPHOST2** later in the scale out process. (Running Unpack to Transfer the Template describes unpack steps.)

For example:

```
ORACLE_HOME/oracle_common/common/bin/pack.sh \
     -domain=DOMAIN_HOME \
     -template=dir/domain_name.jar \
     -managed=true \
     -template_name="DOMAIN"
```

In the preceding example:

- Replace *DOMAIN_HOME* with the full path to the domain home directory.
- Replace *dir* with the full path to a well-known directory where you will create the new template file.

- Replace *domain_name* in the JAR file name with the domain name. This is the name of the template file that you create with the `pack` command. For example: `mydomain.jar`.

> **Note:**
>
> See pack and unpack Command Reference in *Creating WebLogic Domains and Domain Templates* for more information about creating a Managed Server template.

Preparing the New Machine

To prepare the new machine, **machine_2**, verify that **APPHOST2** can access the shared disk where the Oracle home is installed, or install the same software that you installed on **machine_1**.

For example, if you are scaling out an Oracle Fusion Middleware Infrastructure domain, verify that **APPHOST2** can access the Infrastructure Oracle home.

> **Note:**
>
> If you use shared storage, you can reuse the same installation location.

> **Note:**
>
> If you are running a new installation or reusing binaries by means of shared storage, the path location of the new files must match the original machine's path location exactly.

Running Unpack to Transfer the Template

To unpack the template and transfer the *domain_name*.jar file from **APPHOST1** to **APPHOST2**, run the unpack command.

For example:

```
ORACLE_HOME/oracle_common/common/bin/unpack.sh \
    -domain=user_projects/domains/base_domain2 \
    -template=/tmp/domain_name.jar \
    -app_dir=user_projects/applications/base_domain2
```

Starting the Node Manager

You use Node Manager to start Managed Servers (using the Administration Console or Fusion Middleware Control).

To start Node Manager, run:

```
DOMAIN_HOME/bin/startNodeManager.sh &
```

If you use machine scoped Node Manager, see Using Node Manager in *Administering Node Manager for Oracle WebLogic Server* for more on Node Manager start options.

Starting Managed Servers

You start Managed Servers in the Administration Console.

To start Managed Servers:

1. In the left pane of the Console, expand **Environment**, and select **Servers**.
2. In the **Servers** table, click the name of the Managed Server that you moved to the new machine. Select the **Control** tab.
3. Select the check box next to the name of the server(s) you want to start and click **Start** to start the server(s).

> **Note:**
>
> To use WLST commands or Fusion Middleware Control to start Managed Servers, see Starting and Stopping Managed Servers in *Oracle Fusion Middleware Administering Oracle Fusion Middleware*.

Verifying Machine Scale Out

To determine if the machine scale out succeeded, verify that the server status is **RUNNING** (after you use Administration Console to start it).

Configuring Multicast Messaging for Clusters

You configure clusters to use messaging so that they can communicate whether or not services are available and other information.

- About Multicast and Unicast Messaging for Clusters
  Clusters use messaging to broadcast the availability of services, and heartbeats that indicate continued availability. You configure clusters to use Unicast or Multicast messaging.
- Requirements to Configure Multicast Messaging
  Before you configure Multicast messaging, verify that your system meets system and network requirements.
- Configuring Multicast Messaging
  You configure Multicast messaging in the Administration Console by selecting the cluster you want to enable it for then selecting the Multcast Messaging mode.

About Multicast and Unicast Messaging for Clusters

Clusters use messaging to broadcast the availability of services, and heartbeats that indicate continued availability. You configure clusters to use Unicast or Multicast messaging.

- **Multicast** is a simple broadcast technology. Multiple applications subscribe to an IP address and port number then 'listen' for messages. Multicast set up is more complex than Unicast because it needs hardware configuration and support.
- **Unicast** provides TCP-based, reliable, one-to-many communication. It is easier to set up than multicast.

When you create clusters in the Configuration Wizard, Unicast is the default cluster messaging mode. When the number of Managed Servers in a domain increases, Oracle recommends Multicast messaging.

Requirements to Configure Multicast Messaging

Before you configure Multicast messaging, verify that your system meets system and network requirements.

- A configured domain with at least one cluster.
- A hardware configuration set up to support Multicast in your network.
- The IP address and port numbers for Multicast communications in the cluster. A multicast address is an IP address between 224.0.0.0 and 239.255.255.255. (Weblogic uses default value of 239.192.0.0).
- You have run the MulticastTest utility to verify that your environment can support Multicast. The utility debugs Multicast problems; it sends out multicast packets and returns data about how effectively multicast works in your network.

> **Note:**
>
> See Communications In a Cluster in *Administering Clusters for Oracle WebLogic Server* for details on Multicast messaging and cluster configuration.

Configuring Multicast Messaging

You configure Multicast messaging in the Administration Console by selecting the cluster you want to enable it for then selecting the Multcast Messaging mode.

To configure Multicast Messaging:

1. Log in to the Administration Console.
2. Shut down all servers that you want to use for Multicast messaging.
3. In the Domain Structure pane on the left, expand **Environment** and click **Clusters**.
4. Select the cluster name that you want to enable Multicast for.
5. Click the **Configuration** tab then the **Messaging** tab.

6. For Messaging Mode, select **Multicast**. Enter the Multicast Address then the Multicast Port.

   The **Multicast Address** must be unique for each cluster. See Cluster Multicast Address and Port in *Oracle Fusion Middleware Administering Clusters for Oracle WebLogic Server* for guidelines on Multicast addresses.

7. Click **Advanced** to configure these parameters:

Table 6-2 Multicast Advanced Parameters

Parameter	Description
Multicast Send Delay	Amount of time (between 0 and 250) in milliseconds to delay sending message fragments over multicast to avoid OS-level buffer overflow.
Multicast TTL	Number of network hops (between 1 and 255) that a cluster multicast message can travel. If your cluster spans multiple subnets in a WAN, this value must be high enough to ensure that routers don't discard multicast packets before they reach their final destination. This parameter sets the number of network hops a multicast message makes before a router can discard a packet.
Multicast Buffer Size	Multicast socket send/receive buffer size (at least 64 kilobytes).
Idle Periods Until Timeout	Maximum number of periods a cluster member waits before timing out a cluster member.
Enable Data Encryption	Enables multicast data encryption. Only multicast data is encrypted; Multicast header information isn't encrypted.

8. Click **Save**.
9. Restart all servers in the cluster.

The cluster can now use Multicast messaging. When you select **Clusters** in the Domain Structure pane in the Administration Console, **Multicast** appears for **Cluster Messaging Mode**.

7

Using Dynamic Clusters

A **dynamic cluster** is a cluster that contains one or more dynamic servers. A **dynamic server** is a server instance that gets its configuration from a server template. This is in contrast to Managed Servers, which require you to configure each server individually.

- About Dynamic Clusters
  Dynamic clusters consist of server instances that can be dynamically scaled up to meet the resource needs of your application. A dynamic cluster uses a single server template to define configuration for a specified number of generated (dynamic) server instances.
- Why Do You Use Dynamic Clusters?
  With dynamic clusters, you can easily scale up your cluster when you need additional server capacity by simply starting one or more of the preconfigured dynamic server instances. You do not need to manually configure a new server instance and add it to the cluster or perform a system restart.
- How Do Dynamic Clusters Work?
  You set the number of managed server instances that you want, and create or select a template for them.
- Creating Dynamic Clusters in a High Availability Topology
  You can create dynamic clusters to scale out a high availability topology.
- Expanding or Reducing Dynamic Clusters
  When you create a dynamic cluster, WebLogic Server generates the number of dynamic servers you specify. Before you decide upon the number of server instances, ensure you have the performance capacity to handle the desired number.

About Dynamic Clusters

Dynamic clusters consist of server instances that can be dynamically scaled up to meet the resource needs of your application. A dynamic cluster uses a single server template to define configuration for a specified number of generated (dynamic) server instances.

When you create a dynamic cluster, the dynamic servers are preconfigured and automatically generated for you, enabling you to easily scale up the number of server instances in your dynamic cluster when you need additional server capacity. You can simply start the dynamic servers without having to first manually configure and add them to the cluster.

If you need additional server instances on top of the number you originally specified, you can increase the maximum number of servers instances (dynamic) in the dynamic cluster configuration or manually add configured server instances to the dynamic cluster. A dynamic cluster that contains both dynamic and configured server instances is called a mixed cluster.

The following table defines terminology associated with dynamic clusters:

Term	Definition
dynamic cluster	A cluster that contains one or more generated (dynamic) server instances that are based on a single shared server template.
configured cluster	A cluster in which you manually configure and add each server instance.

Term	Definition
dynamic server	A server instance that is generated by WebLogic Server when creating a dynamic cluster. Configuration is based on a shared server template.
configured server	A server instance for which you manually configure attributes.
mixed cluster	A cluster that contains both dynamic and configured server instances.
server template	A prototype server definition that contains common, non-default settings and attributes that can be assigned to a set of server instances, which then inherit the template configuration. For dynamic clusters, the server template is used to generate the dynamic servers. See Server Templates in *Understanding Domain Configuration for Oracle WebLogic Server*.

For more information about dynamic clusters, see Dynamic Clusters in *Administering Clusters for Oracle WebLogic Server*.

Why Do You Use Dynamic Clusters?

With dynamic clusters, you can easily scale up your cluster when you need additional server capacity by simply starting one or more of the preconfigured dynamic server instances. You do not need to manually configure a new server instance and add it to the cluster or perform a system restart.

How Do Dynamic Clusters Work?

You set the number of managed server instances that you want, and create or select a template for them.

- Creating and Configuring Dynamic Clusters
  To create a dynamic cluster, you need to complete three steps in the Configuration Wizard.
- Using Server Templates
  Server templates define common configuration attributes that a set of server instances share. Dynamic clusters use server templates for dynamic server configuration.
- Calculating Server-Specific Attributes
  You cannot configure individual dynamic server instances or override server template values at the dynamic server level when using a dynamic cluster. Server-specific attributes, such as server name, machines, and listen ports, must be calculated using the values you set when creating the dynamic cluster.

Creating and Configuring Dynamic Clusters

To create a dynamic cluster, you need to complete three steps in the Configuration Wizard.

- Specify the number of server instances you anticipate needing at peak load.
- Create or select the server template that you want to base server configuration on.
- Define how WebLogic Server should calculate server-specific attributes.

WebLogic Server then generates the number of dynamic server instances you specified and applies the calculated attribute values to each dynamic server instance.

> **Note:**
>
> Ensure you have the performance capacity to handle the maximum number of server instances you specify in the dynamic cluster configuration. For design and deployment best practices when creating a cluster, see Clustering Best Practices.

Using Server Templates

Server templates define common configuration attributes that a set of server instances share. Dynamic clusters use server templates for dynamic server configuration.

See Server Templates in *Understanding Domain Configuration for Oracle WebLogic Server*.

Calculating Server-Specific Attributes

You cannot configure individual dynamic server instances or override server template values at the dynamic server level when using a dynamic cluster. Server-specific attributes, such as server name, machines, and listen ports, must be calculated using the values you set when creating the dynamic cluster.

> **Note:**
>
> You must set a unique Listen Address value for the Managed Server instance that will host the JTA Transaction Recovery service. Otherwise, the migration fails.

WebLogic Server calculates and applies these server-specific attributes using the dynamic server instance ID:

- Server name
- (Optional) Listen ports (clear text and SSL)
- (Optional) Network access point listen ports
- (Optional) Machines or virtual machines
- Calculating Server Names
  The **Server Name Prefix** controls the calculated server name.
- Calculating Listen Ports
  Calculating Listen Ports specifies whether listen ports for the server are calculated.
- Calculating Machine Names
  The **Calculated Machine Names** and **Machine Name Match Expression** control how server instances in a dynamic cluster are assigned to a machine.

Calculating Server Names

The **Server Name Prefix** controls the calculated server name.

Server names are the prefix that you enter, followed by the index number. For example, if the prefix is set to `dyn-server-`, then the dynamic servers have the names `dyn-server-1`, `dyn-server-2`, and so on for the number of server instances you specified.

Calculating Listen Ports

Calculating Listen Ports specifies whether listen ports for the server are calculated.

If you do not calculate listen ports when creating your dynamic cluster, WebLogic Server uses the value in the server template. If you don't define listen ports in the dynamic cluster configuration or server template, WebLogic Server uses the default value.

If you define a base listen port for your dynamic cluster in the server template or the dynamic cluster configuration, the listen port value for the first dynamic server instance is the base listen port incremented by one. For each additional dynamic server instance, the listen port value increments by one. If the default base listen port is used, WebLogic Server increments the *hundreds* digit by one and continues from there for each dynamic server instance.

Calculating Machine Names

The **Calculated Machine Names** and **Machine Name Match Expression** control how server instances in a dynamic cluster are assigned to a machine.

If you select the **Calculated Machine Names** option, you can use the **Machine Name Match Expression** to choose the set of machines used for the dynamic servers. If you don't set **Machine Name Match Expression**, then *all* machines in the domain are selected. Assignments are made using a round robin algorithm.

The following table shows examples of machine assignments in a dynamic cluster.

Table 7-1 Calculating Machine Names

Machines in Domain	Machine Name Match Expression Configuration	Dynamic Server Machine Assignments
M1, M2	Not set	dyn-server-1: M1 dyn-server-2: M2 dyn-server-3: M1 ...
Ma1, Ma2, Mb1, Mb2	Ma1, Mb*	dyn-server-1: Ma1 dyn-server-2: Mb1 dyn-server-3: Mb2 dyn-server-4: Ma1 ...

Creating Dynamic Clusters in a High Availability Topology

You can create dynamic clusters to scale out a high availability topology.

Before you create dynamic clusters, verify the following:

- Oracle Fusion Middleware Infrastructure and the product software are installed.
- Product schemas are created in the database.

To create dynamic clusters for a high availability topology:

1. Go to the `bin` directory.

   On UNIX operating systems:

   ```
   ORACLE_HOME/oracle_common/common/bin
   ```

   On Windows operating systems:

   ```
   ORACLE_HOME\oracle_common\common\bin
   ```

   where *ORACLE_HOME* is your 12*c* (12.2.1.3.0) Oracle home.

2. Launch the Configuration Wizard:

   On UNIX operating systems:

   ```
   ./config.sh
   ```

   On Windows operating systems:

   ```
   config.cmd
   ```

3. On the **Configuration Type** screen, select **Create a new domain**
4. On the **Templates** screen, enter a name for the template you are creating. Select **Next**. Continue with the Configuration Wizards screens to follow the typical steps to create a cluster, until you reach the **Managed Servers** screen.

   If you need information on the steps to create a cluster starting with the **Application Location** screen (the screen after **Templates** in the Configuration Wizard), see your product's installation guide.

5. (Optional) When you reach the **Managed Servers** screen, you can delete *static* Managed Servers if you want to create a domain with dynamic Managed Servers only. To do this, select the static Managed Servers that you don't want in the domain then select **Delete**. (Don't select the Managed Servers that were already listed.) Click **Next**.
6. On the **Clusters** screen, select **Add**. Enter a name for the cluster and values for **Frontend HTTP Port** and **Frontend HTTPS Port**. Select *product*`-DYN-CLUSTER` in the **Dynamic Server Groups** drop down menu. Select **Unspecified** if a dynamic server group is not listed.
7. The **Server Templates** screen shows templates available for the domain. In the Cluster drop down menu, select the cluster that each server template belongs to.
8. In the **Dynamic Servers** screen, you customize the cluster you just created. Enter **2** as the **Maximum Dynamic Server Count** value. This is the typical number of servers for your first cluster.
9. Select **Calculated Machine Names** and enter a value for **Machine Name Match Expression** to control *how* server instances in a dynamic cluster are assigned to a machine.
   - You must select **Calculated Machine Names** to assign dynamic servers to a specific machine.
   - To choose the set of machines that dynamic servers use, enter a value for **Machine Name Match Expression**. If you *don't* enter a value for **Machine Name Match**

Expression, *all* machines in the domain are selected and a round robin method assigns machines.
If you enter a name, each specified value matches a machine name exactly. Enter a trailing asterisk (*) suffix to match multiple machine names, for example, `SOAHOST*` or `WCPHOST*`

If the **Machine Name Match Expression** is a tag, each specified value matches all machines that have those tag values.

10. Select **Calculated Listen Ports**.
11. Select **Dynamic Cluster**.
12. Click **Next**.

    The **Assign Servers to Clusters** screen opens. In remaining Configuration Wizard screens, you continue to create a domain as you would create a typical domain. See Assigning Managed Servers to the Cluster for information on this screen.

Expanding or Reducing Dynamic Clusters

When you create a dynamic cluster, WebLogic Server generates the number of dynamic servers you specify. Before you decide upon the number of server instances, ensure you have the performance capacity to handle the desired number.

The dynamic server instances are based on the configuration you specified in the server template and calculated attributes.

- To expand your cluster, start any number of the preconfigured dynamic servers.
- To shrink your dynamic cluster, shut down the excess number of dynamic servers.

If you need additional server capacity on top of the number of server instances you originally specified, increase the maximum number of dynamic servers in the dynamic cluster configuration. To reduce the number of server instances in the dynamic cluster, decrease the value of the maximum number of dynamic servers attribute. Before lowering this value, shut down the server instances you plan to remove.

8

JMS and JTA High Availability

To configure Java Message Service (JMS) and Java Transaction API (JTA) services for high availability, you deploy them to migratable targets that can migrate from one server in a cluster to another server.

- About JMS and JTA Services for High Availability
  Java Message Service (JMS) is an application program interface (API) that supports the formal communication known as *messaging* between computers in a network.
- About Migratable Targets for JMS and JTA Services
  To configure JMS and JTA services for high availability, you deploy them to a **migratable target**, a special target that can migrate from one server in a cluster to another.
- Configuring Migratable Targets for JMS and JTA High Availability
  To configure a migratable target, you specify servers that can host a target; only one server can host a migratable target at any one time. You also set the host you prefer for services and back up servers if the preferred host fails.
- User-Preferred Servers and Candidate Servers
  When you deploy a JMS service to a migratable target, you can select a user-preferred server target to host the service. You can also specify **constrained candidate servers** (CCS) that can host a service if the user-preferred server fails.
- Using File Persistence
  Oracle recommends storing the JMS and JTA data in the database for higher reliability, storage in cloud environments, and better consistency in disaster recovery scenarios instead of using file persistence. For example, use the JDBC Store and TLOG-in-DB features for JMS and JTA respectively. If you choose to use a file system, you must use a shared file system for high availability.
- Using File Stores on NFS
  If you store JMS messages and transaction logs on an NFS-mounted directory, Oracle strongly recommends that you verify server restart behavior after an abrupt machine failure. Depending on the NFS implementation, different issues can arise after a failover/restart.
- Configuring WLS JMS with a Database Persistent Store
  You can change WLS JMS configuration from a file-based persistent store (default configuration) to a database persistent store.
- Configuring Database Stores to Persist Transaction Logs
  After you confirm that your setup has a standard Oracle Fusion Middleware installation, you can configure JDBC Transaction Logs (TLOG) Stores.
- Using the Config Wizard for configuring Automatic Service Migration and JDBC Persistent stores for FMW components
  You can use the HA Options screen in the Configuration Wizard to automate the JDBC store persistence and configure service migration. This screen appears for the first time when you create a Fusion Middleware cluster that may use Automatic Service Migration, persistent stores, or both, and all subsequent clusters that are added to the domain by using the Configuration Wizard, automatically apply the selected HA options.

About JMS and JTA Services for High Availability

Java Message Service (JMS) is an application program interface (API) that supports the formal communication known as *messaging* between computers in a network.

Java Transaction API (JTA) specifies standard Java interfaces between a transaction manager and parties involved in a distributed transaction system: the resource manager, the application server, and the transactional applications.

In WebLogic JMS, a message is available only if its host JMS server for the destination is running. If a message is in a central persistent store, the only JMS server that can access the message is the server that originally stored the message. WebLogic has features to restart and/or migrate a JMS server automatically after failures. It also has features for clustering (distributing) a destination across multiple JMS servers within the same cluster.

You automatically restart and / or migrate (fail over) JMS Servers using either Whole Server Migration or Automatic Service Migration.

> **Note:**
>
> For more on working with JMS or JTA, see:
>
> - Configuring WebLogic JMS Clustering in *Oracle Fusion Middleware Administering JMS Resources for Oracle WebLogic Server*
> - Interoperating with Oracle AQ JMS in *Oracle Fusion Middleware Administering JMS Resources for Oracle WebLogic Server*
> - Configuring JTA in *Developing JTA Applications for Oracle WebLogic Server*.
> - Configure Domain JTA Options and Configure Cluster JTA Options in *Administration Console Online Help*.

> **Note:**
>
> For more on Whole Server Migration, see Whole Server Migration .

About Migratable Targets for JMS and JTA Services

To configure JMS and JTA services for high availability, you deploy them to a **migratable target**, a special target that can migrate from one server in a cluster to another.

A migratable target groups migratable services that should move together. When a migratable target migrates, all services that it hosts also migrate.

A migratable target specifies a set of servers that can host a target. Only one server can host a migratable target at any one time. It can also specify:

- A user-preferred host for services
- An ordered list of backup servers if a preferred server fails

After you configure a service to use a migratable target, it is independent from the server member that currently hosts it. For example, if you configure a JMS server with a deployed JMS queue to use a migratable target, the queue is independent of when a specific server member is available. The queue is always available when any server in the cluster hosts the migratable target.

You can migrate pinned migratable services manually from one server to another in the cluster if 1) a server fails, or 2) as part of scheduled maintenance. If you *do not* configure a migratable target in the cluster, migratable services can migrate to any server in the cluster.

See Configuring Migratable Targets for JMS and JTA High Availability to configure migratable targets.

> **Note:**
>
> For more on administering JMS, see the following topics in *Oracle Fusion Middleware Administering JMS Resources for Oracle WebLogic Server*:
>
> - High Availability Best Practices
> - Interoperating with Oracle AQ JMS

Configuring Migratable Targets for JMS and JTA High Availability

To configure a migratable target, you specify servers that can host a target; only one server can host a migratable target at any one time. You also set the host you prefer for services and back up servers if the preferred host fails.

To configure migratable targets, see these topics in Administration Console Online Help:

- Configure Migratable Targets for JMS-Related Services
- Configure Migratable Targets for JTA Transaction Recovery Service

User-Preferred Servers and Candidate Servers

When you deploy a JMS service to a migratable target, you can select a user-preferred server target to host the service. You can also specify **constrained candidate servers** (CCS) that can host a service if the user-preferred server fails.

If a migratable target doesn't specify a CCS, you can migrate the JMS server to any available server in the cluster.

You can create separate migratable targets for JMS services so that you can always keep each service running on a different server in the cluster, if necessary. Conversely, you can configure the same selection of servers as the CCSs for both JTA and JMS, to ensure that services stay co-located on the same server in the cluster.

Using File Persistence

Oracle recommends storing the JMS and JTA data in the database for higher reliability, storage in cloud environments, and better consistency in disaster recovery scenarios instead of using file persistence. For example, use the JDBC Store and TLOG-in-DB features for JMS and JTA respectively. If you choose to use a file system, you must use a shared file system for high availability.

WebLogic supplies multiple types of file stores which have multiple purposes as follows:

- Each WebLogic Server has a default file store. However, default file stores should not be used for storing critical data such as JMS messages, JTA transactions, and EJB timers. You should use JDBC stores, TLOG-in-DB, and database stored timers instead. An example of non-critical data that is stored in a default store is application life-cycle state, such as whether a particular application has been administratively paused. If there is no critical data in a default file store, it is safe to delete such stores in the event of a catastrophic corruption as this mitigates the risk of disabling file locking for the default file store.
- WebLogic JMS paging stores are active if JMS has a large message backlog. The data in JMS paging files is not reloaded when the server is not running, and so the files can be safely deleted when a WebLogic Server is not running. The corresponding persistent messages are simultaneously stored in default file stores, custom file stores, or custom JDBC stores.
- WebLogic diagnostic stores contain non-critical diagnostic data. They are run within a buffering mode that allows for very high performance in order to minimize the overhead of diagnostics, but this increases the risk of corruption after a failure. If such files become corrupt, then it is safe for WebLogic Servers to reboot, and it is also safe to delete the files.

See Tuning the WebLogic Persistent Store section in *Tuning Performance of Oracle WebLogic Server*.

Using File Stores on NFS

If you store JMS messages and transaction logs on an NFS-mounted directory, Oracle strongly recommends that you verify server restart behavior after an abrupt machine failure. Depending on the NFS implementation, different issues can arise after a failover/restart.

- Prerequisites for Disabling File Locking
- Disabling File Locking for all Stores Using a System Property
- Verifying Server Restart Behavior
  To verify server restart behavior, abruptly shut down the node that hosts WebLogic servers while the servers are running.
- Disabling File Locking for the Default File Store
  If WebLogic Server doesn't restart after abrupt machine failure and server log files show the NFS system doesn't release locks on file stores, you can disable file locking.

- Disabling File Locking for a Custom File Store
  If WebLogic Server doesn't restart after abrupt machine failure and server log files show the NFS system doesn't release locks on custom file stores, you can disable file locking.
- Disabling File Locking for a JMS Paging File Store
  If WebLogic Server doesn't restart after abrupt machine failure and server log files show the NFS system doesn't release locks on JMS paging file stores, you can disable file locking.
- Disabling File Locking for Diagnostics File Stores
  If WebLogic Server doesn't restart after abrupt machine failure and server log files show the NFS system doesn't release locks on diagnostics paging file stores, you can disable file locking.

Prerequisites for Disabling File Locking

All file stores are locked by default. The WebLogic file locking feature is designed to help prevent severe file corruptions that can occur in concurrency scenarios. Perform the following prerequisite tasks to mitigate the risk of disabling file locks:

- If the server using the file store is configured for server migration, always configure the database-based cluster leasing option instead of the default consensus leasing. This enforces additional locking mechanisms using database tables and prevents automated concurrent restart of more than one instance of a particular WebLogic Server.
- Avoid disabling file locks on a file store that is using Automatic Service Migration (ASM).
  - ASM requires file store locking to work safely and is activated in the following scenarios:
    1. A custom file store target is set to a Migratable Target and the Migratable Target is configured with a Migration Policy other than 'manual' (the default).
    2. A custom file store target is set to a WebLogic cluster when the store is configured with a Migration Policy other than 'Off' (the default).
    3. A WebLogic Server is configured with a JTA Migratable Target with a Migration Policy value other than ‘manual’ (the default), as this in turn can lead to default file store migrations.
  - If both ASM and disabling file locks are required, store your critical data in database stores instead of file stores to avoid the risk of file corruptions. For example, use a custom JDBC store instead of a file store and configure JTA to use a JDBC TLOG store instead of each WebLogic Server’s default file store.
- Additional procedural precautions must be implemented to avoid any human error and to ensure that only one instance of a server is manually started at any given point in time. Similarly, precautions must be taken to ensure that no two domains have a store with the same name that references the same directory.

Disabling File Locking for all Stores Using a System Property

Starting from WebLogic Server 14.1.2 release, you can specify a Java system property on the `weblogic.Server` command line or start script of the JVM to disable locking on all of its file stores including default, paging, diagnostic, and custom file stores as shown below:

```
"-Dweblogic.store.file.LockEnabled=false"
```

Verifying Server Restart Behavior

To verify server restart behavior, abruptly shut down the node that hosts WebLogic servers while the servers are running.

- If you *configured the server for server migration,* it should start automatically in failover mode after the failover period.
- If you *did not* configure the server for server migration, you can manually restart the WebLogic Server on the same host after the node completely reboots.

If WebLogic Server doesn't restart after abrupt machine failure, review server log files to verify whether or not it is due to an I/O exception similar to the following:

```
<MMM dd, yyyy hh:mm:ss a z> <Error> <Store> <BEA-280061> <The persistent
store "_WLS_server_1" could not be deployed:
weblogic.store.PersistentStoreException: java.io.IOException:
[Store:280021]There was an error while opening the file store file
"_WLS_SERVER_1000000.DAT"
weblogic.store.PersistentStoreException: java.io.IOException:
[Store:280021]There was an error while opening the file store file
"_WLS_SERVER_1000000.DAT"
at weblogic.store.io.file.Heap.open(Heap.java:168)
at weblogic.store.io.file.FileStoreIO.open(FileStoreIO.java:88)
...
java.io.IOException: Error from fcntl() for file locking, Resource
temporarily unavailable, errno=11
```

This error occurs when the NFSv3 system doesn't release locks on file stores. WebLogic Server maintains locks on files that store JMS data and transaction logs to prevent data corruption that can occur if you accidentally start two instances of the same Managed Server. Because the NFSv3 storage device doesn't track lock owners, NFS holds the lock indefinitely if a lock owner fails. As a result, after abrupt machine failure followed by a restart, subsequent attempts by WebLogic Server to acquire locks may fail.

How you resolve this error depends on your NFS environment: (See *Oracle Fusion Middleware Release Notes* for updates on this topic.)

- **For NFSv4 environments**, you can set a tuning parameter on the NAS server to release locks within the approximate time required to complete server migration; you don't need to follow procedures in this section. See your storage vendor's documentation for information on locking files stored in NFS-mounted directories on the storage device, and test the results.
- **For NFSv3 environments**, the following sections describe how to disable WebLogic file locking mechanisms for: the default file store, a custom file store, a JMS paging file store, a diagnostics file store.

> **WARNING:**
>
> NFSv3 file locking prevents severe file corruptions that occur if more than one Managed Server writes to the same file store at any point in time.
>
> If you disable NFSv3 file locking, you must implement administrative procedures / policies to ensure that only one Managed Server writes to a specific file store. Corruption can occur with two Managed Servers in the same cluster or different clusters, on the same node or different nodes, or on the same domain or different domains.
>
> Your policies could include: never copy a domain, never force a unique naming scheme of WLS-configured objects (servers, stores), each domain must have its own storage directory, no two domains can have a store with the same name that references the same directory.
>
> When you use a file store, always configure the database-based leasing option for server migration. This option enforces additional locking mechanisms using database tables and prevents automated restart of more than one instance of a particular Managed Server.

Disabling File Locking for the Default File Store

If WebLogic Server doesn't restart after abrupt machine failure and server log files show the NFS system doesn't release locks on file stores, you can disable file locking.

To disable file locking for the default file store using the Administration Console:

1. If necessary, click **Lock & Edit** in the Change Center.
2. In the **Domain Structure** tree, expand the **Environment** node and select **Servers**.
3. In the **Summary of Servers** list, select the server you want to modify.
4. Select the **Configuration > Services** tab.
5. Scroll down to the **Default Store** section and click **Advanced**.
6. Scroll down and deselect the **Enable File Locking** check box.
7. Click **Save**. If necessary, click **Activate Changes** in the Change Center.
8. **Restart** the server you modified for the changes to take effect.

The resulting `config.xml` entry looks like the following:

```
<server>
 <name>examplesServer</name>
 ...
 <default-file-store>
 <synchronous-write-policy>Direct-Write</synchronous-write-policy>
 <io-buffer-size>-1</io-buffer-size>
 <max-file-size>1342177280</max-file-size>
 <block-size>-1</block-size>
 <initial-size>0</initial-size>
 <file-locking-enabled>false</file-locking-enabled>
 </default-file-store>
 </server>
```

Disabling File Locking for a Custom File Store

If WebLogic Server doesn't restart after abrupt machine failure and server log files show the NFS system doesn't release locks on custom file stores, you can disable file locking.

To disable file locking for a custom file store using the Administration Console:

1. If necessary, click **Lock & Edit** in the Change Center to get an Edit lock for the domain.
2. In the **Domain Structure** tree, expand the **Services** node. Select **Persistent Stores**.
3. In the **Summary of Persistent Stores** list, select the custom file store you want to modify.
4. On the **Configuration** tab for the custom file store, click **Advanced**.
5. Scroll down and deselect the **Enable File Locking** check box.
6. Click **Save**. If necessary, click **Activate Changes** in the Change Center.
7. If the custom file store was in use, you must restart the server for changes to take effect.

The `config.xml` entry looks like the following example:

```
<file-store>
 <name>CustomFileStore-0</name>
 <directory>C:\custom-file-store</directory>
 <synchronous-write-policy>Direct-Write</synchronous-write-policy>
 <io-buffer-size>-1</io-buffer-size>
 <max-file-size>1342177280</max-file-size>
 <block-size>-1</block-size>
 <initial-size>0</initial-size>
 <file-locking-enabled>false</file-locking-enabled>
 <target>examplesServer</target>
 </file-store>
```

Disabling File Locking for a JMS Paging File Store

If WebLogic Server doesn't restart after abrupt machine failure and server log files show the NFS system doesn't release locks on JMS paging file stores, you can disable file locking.

To disable file locking for a JMS paging file store using the Administration Console:

1. If necessary, click **Lock & Edit** in the Change Center to get an Edit lock for the domain.
2. In the **Domain Structure** tree, expand the **Services** node, expand the **Messaging** node, and select **JMS Servers**.
3. In the **Summary of JMS Servers** list, select a JMS server to modify.
4. On the **Configuration > General** tab for the JMS Server, scroll down. Deselect the **Paging File Locking Enabled** check box.
5. Click **Save**. If necessary, click **Activate Changes** in the Change Center.
6. **Restart** the server you modified for changes to take effect.

The `config.xml` file entry looks like the following example:

```
<jms-server>
<name>examplesJMSServer</name>
<target>examplesServer</target>
<persistent-store>exampleJDBCStore</persistent-store>
...
<paging-file-locking-enabled>false</paging-file-locking-enabled>
...
</jms-server>
```

Disabling File Locking for Diagnostics File Stores

If WebLogic Server doesn't restart after abrupt machine failure and server log files show the NFS system doesn't release locks on diagnostics paging file stores, you can disable file locking.

To disable file locking for a Diagnostics file store using the Administration Console:

1. If necessary, click **Lock & Edit** in the Change Center to get an Edit lock for the domain.
2. In the **Domain Structure** tree, expand the **Diagnostics** node. Select **Archives**.
3. In the **Summary of Diagnostic Archives** list, select the server name of the archive that you want to modify.
4. On the **Settings for [server_name]** page, deselect the **Diagnostic Store File Locking Enabled** check box.
5. Click **Save**. If necessary, click **Activate Changes** in the Change Center.
6. **Restart** the server you modified for the changes to take effect.

The resulting `config.xml` file looks like this:

```
<server>
<name>examplesServer</name>
...
<server-diagnostic-config>
<diagnostic-store-dir>data/store/diagnostics</diagnostic-store-dir>
<diagnostic-store-file-locking-enabled>false</diagnostic-store-file-locking-
enabled>
<diagnostic-data-archive-type>FileStoreArchive</diagnostic-data-archive-type>
<data-retirement-enabled>true</data-retirement-enabled>
<preferred-store-size-limit>100</preferred-store-size-limit>
<store-size-check-period>1</store-size-check-period>
</server-diagnostic-config>
</server>
```

> **Note:**
>
> See Configure JMS Servers and Persistent Stores in *Oracle Fusion Middleware Administering JMS Resources for Oracle WebLogic Server*.

Configuring WLS JMS with a Database Persistent Store

You can change WLS JMS configuration from a file-based persistent store (default configuration) to a database persistent store.

- About the Persistent Store
  The **persistent store** is a built-in storage solution for WebLogic Server subsystems and services that require persistence. For example, it can store persistent JMS messages.
- Prerequisites for Configuring WLS JMS with a Database Persistent Store
  To configure WLS JMS with database persistent stores, verify that your setup meets specific requirements.
- Switching WLS JMS File-Based Persistent Stores to Database Persistent Store
  You can swap JMS servers from file-based to database persistent stores.

About the Persistent Store

The **persistent store** is a built-in storage solution for WebLogic Server subsystems and services that require persistence. For example, it can store persistent JMS messages.

The persistent store supports persistence to a file-based store or to a JDBC-accessible store in a database. For information on the persistent store, see The WebLogic Persistent Store in *Administering the WebLogic Server Persistent Store*.

For information on typical tasks to monitor, control, and configure WebLogic messaging components, see WebLogic Server Messaging in *Administering Oracle WebLogic Server with Fusion Middleware Control*.

Prerequisites for Configuring WLS JMS with a Database Persistent Store

To configure WLS JMS with database persistent stores, verify that your setup meets specific requirements.

Your setup must meet these requirements:

- An Oracle Fusion Middleware installation with at least one cluster and one or more JMS servers
- JMS servers that use file persistent stores, the default configuration.

Switching WLS JMS File-Based Persistent Stores to Database Persistent Store

You can swap JMS servers from file-based to database persistent stores.

You must follow steps in this procedure for each JMS server that you must configure to use database persistent stores.

1. Create a JDBC store. See Using a JDBC Store in *Oracle Fusion Middleware Administering Server Environments for Oracle WebLogic Server*.

> **Note:**
>
> You must specify a prefix to uniquely name the database table for the JDBC store.

2. Associate the JDBC store with the JMS server:
   a. In the Weblogic Server Administration Console, go to **Services**->**Messaging**->**JMS Servers**.
   b. Verify that there are no pending messages in this server. In the Control tab, stop production and insertion of messages for all destinations and wait for any remaining messages to drain.
   c. Select the General Configuration tab. Under Persistent Store, select the new JDBC store then click **Save**.

The JMS server starts using the database persistent store.

Configuring Database Stores to Persist Transaction Logs

After you confirm that your setup has a standard Oracle Fusion Middleware installation, you can configure JDBC Transaction Logs (TLOG) Stores.

- Requirements for Configuring JDBC TLOG Stores
  You must have a standard Oracle Fusion Middleware installation before you configure a JDBC Transaction Logs (TLOG) Store.
- Configuring JDBC TLOG Stores
  There are a few guidelines to follow when you configure JDBC TLOG Stores for Managed Servers in a cluster.

Requirements for Configuring JDBC TLOG Stores

You must have a standard Oracle Fusion Middleware installation before you configure a JDBC Transaction Logs (TLOG) Store.

Post installation, the TLOG Store is configured in the file system. In some instances, Oracle recommends that you configure TLOGs to store in the database. To configure JDBC TLOGs to stored to a database store, see Using a JDBC TLOG Store in *Administering the WebLogic Server Persistent Store*.

Configuring JDBC TLOG Stores

There are a few guidelines to follow when you configure JDBC TLOG Stores for Managed Servers in a cluster.

When you configure JDBC TLOG Stores:

- You must repeat the procedure for each Managed Server in the cluster.
- Use the Managed Server name as a prefix to create a unique TLOG store name for each Managed Server.
- Verify that the data source that you created for the persistent store targets the cluster for a high availability setup.

When you finish the configuration, TLOGs are directed to the configured database-based persistent store.

> **Note:**
>
> When you add a new Managed Server to a cluster by scaling up or scaling out, you must also create the corresponding JDBC TLOG Store for the new Managed Server.

Using the Config Wizard for configuring Automatic Service Migration and JDBC Persistent stores for FMW components

You can use the HA Options screen in the Configuration Wizard to automate the JDBC store persistence and configure service migration. This screen appears for the first time when you create a Fusion Middleware cluster that may use Automatic Service Migration, persistent stores, or both, and all subsequent clusters that are added to the domain by using the Configuration Wizard, automatically apply the selected HA options.

If you select the **Enable Automatic Service Migration** option, it configures migrtable target definitions that are required for automatic service migration. You can either select database or consensus leasing option. If you select **Database Leasing**, the leasing datasource is also configured.

In the same screen, use the options **JTA Transaction Log Persistence** and **JMS Server Persistence** to configure JDBC stores automatically. Fusion Middleware component templates automatically define the JDBC persistent stores for JMS Servers and Transaction Logs.

For information about what to select in the High Availability options screen during the domain creation process, see Configuring High Availability Options .

9

Administration Server High Availability

The Administration Server plays a unique role in domains. To set up high availability, you configure the Administration Server on a virtual host.

- Administration Server Role
  The Administration Server is the central control entity for configuring the entire domain, and manage and monitor all domain resources. It maintains domain configuration documents and distributes changes in them to Managed Servers.
- Role of Node Manager
  For each WebLogic Server domain you create, a *per domain* Node Manager configuration is created by default. This Node Manager comes complete with security credentials, a properties file, domain registration, and start scripts.
- Administration Server High Availability Topology
  An Administration Server can be active on one host at any one time. To set up high availability, you configure the Administration Server on a virtual host so that if the machine that it runs on fails, you can fail it over to another host in the domain.
- Configuring Administration Server High Availability
  In a highly available Administration Server environment, both hosts must have access to shared storage because the domain directory is maintained in shared storage.
- Failing Over or Failing Back Administration Server
  You fail over or fail back the Administration Server after host failure.

Administration Server Role

The Administration Server is the central control entity for configuring the entire domain, and manage and monitor all domain resources. It maintains domain configuration documents and distributes changes in them to Managed Servers.

Each domain requires one server that acts as the Administration Server. For more information on the Administration Server and Node Manager, see the following topics:

Table 9-1 Administration Server and Node Manager Topics

For information on...	See this topic...
Starting and Stopping the Administration Server	Starting and Stopping Administration Server in *Administering Oracle Fusion Middleware*
Configuring virtual hosting	Configuring Virtual Hosting in *Administering Server Environments for Oracle WebLogic Server*
Using Node Manager	In *Administering Node Manager for Oracle WebLogic Server*: • Node Manager and System Crash Recovery • How Node Manager Works in a WLS Environment • Node Manager Configuration and Log Files

- Administration Server Failure and Restart
  Administration Server failure doesn't affect how Managed Servers in a domain operate.

- Shared Storage and Administration Server High Availability
  With shared storage, a backup host can access the same artifacts (Oracle binaries, configuration files, domain directory, and data) that an active host can.

Administration Server Failure and Restart

Administration Server failure doesn't affect how Managed Servers in a domain operate.

If an Administration Server fails due to a hardware or software failure on its host computer, other server instances on the same computer may also be affected.

For more information on Administration Server failure, see Impact of Managed Server Failure in *Administering Oracle Fusion Middleware*.

To restart an Administration Server, see What Happens if the Administration Server Fails? in *Oracle Fusion Middleware Using Clusters for Oracle Server*.

Shared Storage and Administration Server High Availability

With shared storage, a backup host can access the same artifacts (Oracle binaries, configuration files, domain directory, and data) that an active host can.

You configure this access by placing artifacts in storage that all hosts in the highly available Administration Server configuration can access. Shared storage is a network-attached storage (NAS) or storage area network (SAN) device. See Using Shared Storage.

Role of Node Manager

For each WebLogic Server domain you create, a *per domain* Node Manager configuration is created by default. This Node Manager comes complete with security credentials, a properties file, domain registration, and start scripts.

You can configure the scope of Node Manager:

- **per domain** Node Manager is associated with a *domain* to control all servers for the domain on a machine. Default configuration.
- **per host** Node Manager is associated with a specific *machine*, not a domain. One Node Manager process can control server instances in any domain, as long as the server instances reside on the same machine as the Node Manager process. A per host Node Manager must run on each computer that hosts WebLogic Server instances that you want to control with Node Manager, whether the WebLogic Server instances are an Administration Server or Managed Server(s).

> **Note:**
>
> Oracle recommends that you run Node Manager as an operating system service so that it restarts automatically if its host machine restarts.

Node Manager failure doesn't affect any servers running on the machine.

See What is Node Manager? in *Oracle Fusion Middleware Understanding Oracle Fusion Middleware Concepts*.

Administration Server High Availability Topology

An Administration Server can be active on one host at any one time. To set up high availability, you configure the Administration Server on a virtual host so that if the machine that it runs on fails, you can fail it over to another host in the domain.

Administration Server is configured to use a virtual IP to overlap the backup hosts. You configure Administration Server to listen on this virtual IP. The benefit of a virtual host and virtual IP is that you don't need to add a third machine; if failover occurs, you can map the virtual host to a surviving host in the domain by *moving* the virtual IP.

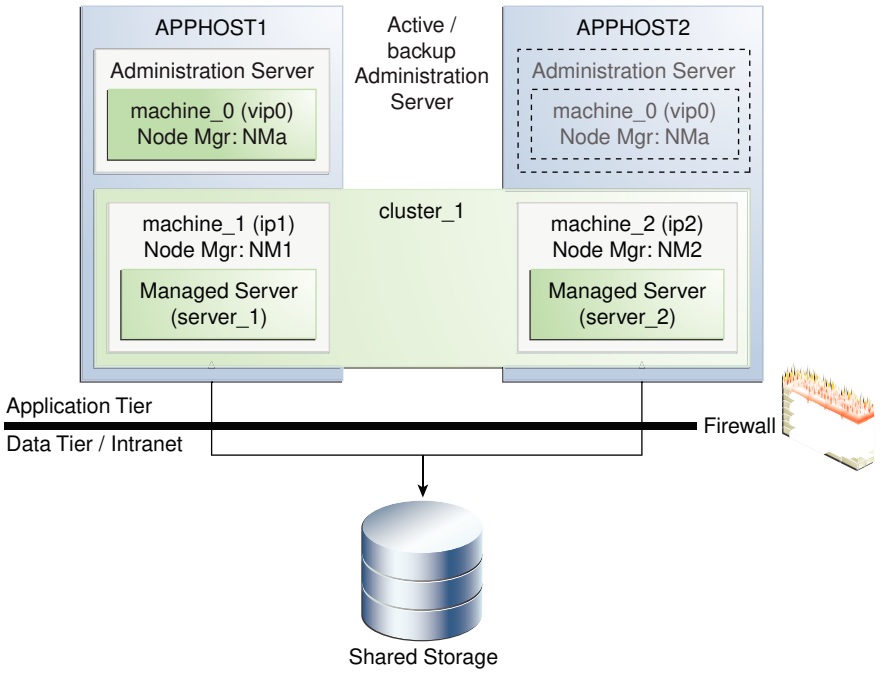
The two hosts share a virtual hostname and a virtual IP. However, only the active host can use this virtual IP at any one time. If the active host fails, the backup host becomes active and you must move (manually) the virtual IP to the new active host. The new active host then services all requests through the virtual IP. (You configure a high availability deployment to listen on this virtual IP.)

Figure 9-1 shows a highly available Administration Server.

In this topology, the Administration Server runs on a virtual host, `APPHOST0`. `APPHOST0` overlaps to `APPHOST1` or `APPHOST2` by means of a virtual IP.

At first, `APPHOST0` maps to `APPHOST1`. However, if the Administration Server fails due to an `APPHOST1` failure, `APPHOST0` fails over to `APPHOST2` by moving the virtual IP.

Figure 9-1 Administration Server High Availability Topology



Configuring Administration Server High Availability

In a highly available Administration Server environment, both hosts must have access to shared storage because the domain directory is maintained in shared storage.

During normal operation, the Administration Server on the active host owns the domain directory in shared storage. If the active host fails, the backup host takes over and restarts the Administration Server from the shared domain directory.

- Administration Server High Availability Requirements
  To configure a highly available Administration Server, your environment must meet certain requirements.
- Configuring the Administration Server for High Availability
  To configure the Administration Server for high availability, you must start with the standard high availability topology that has one cluster (`cluster_1`).

Administration Server High Availability Requirements

To configure a highly available Administration Server, your environment must meet certain requirements.

- Conform to the standard installation topology. See About the Oracle Fusion Middleware Standard HA Topology and Figure 1-1.
- Include two hosts, APPHOST1 and APPHOST2, to implement a WebLogic Server cluster (`cluster_1`). In Configuring the Administration Server for High Availability, IP addresses for APPHOST1 and APPHOST2 are `ip1` and `ip2`, respectively.
- APPHOST1 and APPHOST2 mount a common directory from shared storage and have read/write access rights to the directory. This directory is for installing products and storing domain directories.
- A reserved virtual IP address (`vip0`) to *point* to the host that runs the Administration Server. This floating virtual server IP address is configured dynamically on the host that the Administration Server runs on.
- Node Manager instances to manage the Administration Server and migrate it from the failed host to the designated standby host.

Configuring the Administration Server for High Availability

To configure the Administration Server for high availability, you must start with the standard high availability topology that has one cluster (`cluster_1`).

See About the Oracle Fusion Middleware Standard HA Topology.

To set up a highly available Administration Server, you run the Administration Server on a separate, virtual host (`APPHOST0`). You set up `APPHOST0` so that it maps to one of the existing hosts in the cluster (`APPHOST1` or `APPHOST2`) by configuring a (virtual) server IP for `APPHOST0` on that existing host. If failover occurs, `APPHOST0` fails over by moving its virtual server IP to a surviving host. The domain configuration is on shared storage so that the surviving host can access it.

> **Note:**
>
> There are multiple ways for Administration Server services to accomplish configuration tasks. No matter which method you use, the Administration Server must be running when you change the configuration.

Table 9-2 Host and Node Manager Terms

Term	Description
APPHOST0, machine_0	Virtual machine that the Administration Server runs on
APPHOST1, APPHOST2	Machines that host the application tier.
vip0	Virtual server IP address that the Administration Server listens on
NMa	Per-domain Node Manager that manages the Administration Server that runs on APPHOST0
NM1, NM2	Node Manager instances that run on APPHOST1 and APPHOST2, respectively
ip1, ip2	IP addresses of APPHOST1 and APPHOST2, respectively

To configure the Administration Server for high availability:

1. Configure a virtual server IP address (vip0) on APPHOST1 to represent virtual host APPHOST0.

   See Configuring Virtual Hosting in *Administering Server Environments for Oracle WebLogic Server*.

2. Use an Oracle Fusion Middleware expanded installation procedure to install Oracle Fusion Middleware binaries and configure the domain into a directory on shared storage. Use vip0 as the Administration Server listen address.

3. Create a virtual machine, machine_0 and add the Administration Server to it. machine_0 represents the virtual server host APPHOST0 with the IP address vip0.

4. Create a cluster (cluster_1) that has two Managed Servers, server_1 and server_2, that are assigned to machine_1 and machine_2, respectively.

   - machine_1 represents APPHOST1 and machine_2 represents APPHOST2.
   - server_1 and server_2 are set up to listen on ip1 and ip2, respectively.

5. Scale out the virtual server to APPHOST1 and APPHOST2. See Roadmap for Scaling Out Your Topology. To scale out, you pack the domain in shared storage and unpack it to a local directory on APPHOST1 and APPHOST2.

6. From APPHOST1, start a per-domain Node Manager (NMa) to manage the Administration Server listening on the configured virtual server IP vip0 on APPHOST1. Start this instance of Node Manager from the domain directory in shared storage.

7. On APPHOST1, start a per-domain Node Manager (NM1) to manage server_1 listening on ip1. Start this Node Manager (NM1) from the domain directory that you unpacked to local storage in APPHOST1 in step 5.

8. On APPHOST2, start a per-domain Node Manager (NM2) to manage server_2 listening on ip2. Start this Node Manager (NM2) from the domain directory that you unpacked to local storage in APPHOST2 step 5.
9. Use Node Manager (NMa) to start the Administration Server on APPHOST1.
10. Start Managed Servers server_1 and server_2 using NM1 and NM2, respectively.
11. Verify that the Administration Server and Managed Servers work properly. Connect to the Administration Server using the virtual IP address vip0.

Failing Over or Failing Back Administration Server

You fail over or fail back the Administration Server after host failure.

- Failing Over the Administration Server if Original Host Fails
  You fail over the Administration Server to another host if its original host (APPHOST1) fails. To do this, you configure the virtual IP address on the alternate host (APPHOST2), then start Node Manager and the Administration Server.
- Failing Back the Administration Server to the Original Host
  You fail back the Administration Server to its original host after it restarts.

Failing Over the Administration Server if Original Host Fails

You fail over the Administration Server to another host if its original host (APPHOST1) fails. To do this, you configure the virtual IP address on the alternate host (APPHOST2), then start Node Manager and the Administration Server.

To fail over the Administration Server to APPHOST2 if APPHOST1 fails:

1. Configure vip0 on APPHOST2.
2. Start Node Manager NMa on APPHOST2 from the domain directory in shared storage.
3. Start the Administration Server on APPHOST2 using NMa.
4. Start the Administration Console to verify that the Administration Server is running.

Failing Back the Administration Server to the Original Host

You fail back the Administration Server to its original host after it restarts.

To fail back the Administration Server to APPHOST1 when APPHOST1 comes back online:

1. Stop the Administration Server on APPHOST2 using Node Manager NMa.
2. Remove vip0 from APPHOST2.
3. Stop Node Manager NMa on APPHOST2.
4. Configure vip0 on APPHOST1.
5. Start Node Manager NMa on APPHOST1 using the domain directory in shared storage.
6. Use Node Manager NMa to start the Administration Server on APPHOST1.
7. Start the Administration Console to verify that the Administration Server is running.

Part III
Component Procedures

Part III describes procedures that are unique to certain component products.

This part includes the following topics:

- Configuring High Availability for Oracle Identity Governance Components
  This chapter describes how to design and deploy a high availability environment for Oracle Identity Governance.
- Configuring High Availability for Oracle Access Manager Components
  An introduction to Oracle Access Manager and description of how to design and deploy a high availability environment for Access Manager.
- Configuring High Availability for Oracle Directory Services Components
  This chapter describes configuring Oracle Directory Services products for high availability in an active-active configuration.
- Configuring High Availability for Web Tier Components
- Configuring High Availability for SOA Components
- Configuring High Availability for Oracle WebCenter Components
- Configuring High Availability for Other Components

10

Configuring High Availability for Oracle Identity Governance Components

This chapter describes how to design and deploy a high availability environment for Oracle Identity Governance.

Oracle Identity Governance (OIG) is a user provisioning and administration solution that automates the process of adding, updating, and deleting user accounts from applications and directories. It also improves regulatory compliance by providing granular reports that attest to who has access to what. OIG is available as a stand-alone product or as part of Oracle Identity and Access Management Suite.

For details about OIG, see in Product Overview for Oracle Identity Governance in *Administering Oracle Identity Governance*.

> **Note:**
>
> Oracle Identity Governance and Oracle Identity Manager product name references in the documentation mean the same.

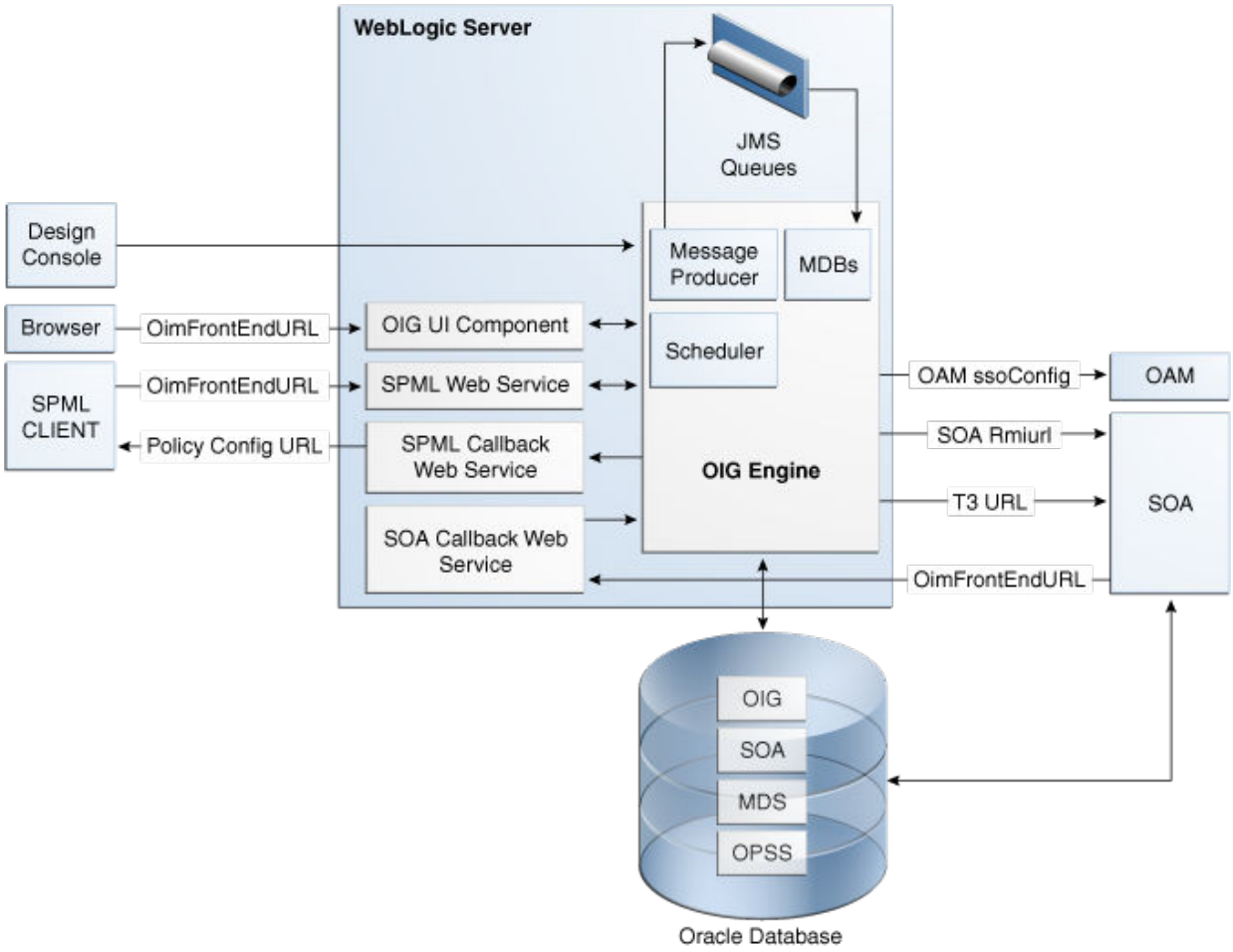
- Oracle Identity Governance Architecture
  Oracle Identity Governance architecture consists of its components, runtime processes, process lifecycle, configuration artifacts, external dependencies, and log files.
- Oracle Identity Governance High Availability Concepts
  The concepts related to Oracle Identity Governance High Availability are OIG high availability architecture, starting and stopping OIG cluster, and cluster-wide configuration changes.
- High Availability Directory Structure Prerequisites
  A high availability deployment requires product installations and files to reside in specific directories. A standard directory structure makes facilitates configuration across nodes and product integration.
- Oracle Identity Governance High Availability Configuration Steps
  Oracle Identity Governance high availability configuration involves setting the prerequisites, configuring the domain, post-installation steps, starting servers, SOA integration, validating managed server instances, and scaling up and scaling out Oracle Identity Governance.

Oracle Identity Governance Architecture

Oracle Identity Governance architecture consists of its components, runtime processes, process lifecycle, configuration artifacts, external dependencies, and log files.

Figure 10-1 shows the Oracle Identity Governance architecture:

Figure 10-1 Oracle Identity Governance Component Architecture



Oracle Identity Governance Component Characteristics

Oracle Identity Manager Server is Oracle's self-contained, standalone identity management solution. It provides User Administration, Workflow and Policy, Password Management, Audit and Compliance Management, User Provisioning and Organization and Role Management functionalities.

Oracle Identity Manager (OIM) is a standard Java EE application that is deployed on WebLogic Server and uses a database to store runtime and configuration data. The

MDS schema contains configuration information; the runtime and user information is stored in the OIM schema.

OIM connects to the SOA Managed Servers over RMI to invoke SOA EJBs.

OIM uses the human workflow module of Oracle SOA Suite to manage its request workflow. OIM connects to SOA using the T3 URL for the SOA server, which is the front end URL for SOA. Oracle recommends using the load balancer or web server URL for clustered SOA servers. When the workflow completes, SOA calls back OIM web services using OIMFrontEndURL. Oracle SOA is deployed along with the OIM.

Several OIM modules use JMS queues. Each queue is processed by a separate Message Driven Bean (MDB), which is also part of the OIM application. Message producers are also part of the OIM application.

OIM uses a Quartz based scheduler for scheduled activities. Various scheduled activities occur in the background, such as disabling users after their end date.

In this release, BI Publisher is not embedded with OIM. However, you can integrate BI Publisher with OIM by following the instructions in Configuring Reports in *Developing and Customizing Applications for Oracle Identity Governance*.

Runtime Processes

Oracle Identity Manager deploys on WebLogic Server as a no-stage application. The OIM server initializes when the WebLogic Server it is deployed on starts up. As part of application initialization, the quartz-based scheduler is also started. Once initialization is done, the system is ready to receive requests from clients.

You must start the Design Console separately as a standalone utility.

Component and Process Lifecycle

Oracle Identity Manager deploys to a WebLogic Server as an externally managed application. By default, WebLogic Server starts, stops, monitors and manages other lifecycle events for the OIM application.

OIM starts after the application server components start. It uses the authenticator which is part of the OIM component mechanism; it starts up before the WebLogic JNDI initializes and the application starts.

OIM uses a Quartz technology-based scheduler that starts the scheduler thread on all WebLogic Server instances. It uses the database as centralized storage for picking and running scheduled activities. If one scheduler instance picks up a job, other instances do not pick up that same job.

You can configure Node Manager to monitor the server process and restart it in case of failure.

The Oracle Enterprise Manager Fusion Middleware Control is used to monitor as well as to modify the configuration of the application.

Starting and Stopping Oracle Identity Governance

You manage OIM lifecycle events with these command line tools and consoles:

- Oracle WebLogic Scripting Tool (WLST)

- WebLogic Server Administration Console
- Oracle Enterprise Manager Fusion Middleware Control
- Oracle WebLogic Node Manager

Configuration Artifacts

The OIM server configuration is stored in the MDS repository at `/db/oim-config.xml`. The `oim-config.xml` file is the main configuration file. Manage OIM configuration using the MBean browser through Oracle Enterprise Manager Fusion Middleware Control or command line MDS utilities. For more information about MDS utilities, see Migrating User Configurable Metadata Files in *Developing and Customizing Applications for Oracle Identity Governance*.

The installer configures JMS out-of-the-box; all necessary JMS queues, connection pools, data sources are configured on WebLogic application servers. These queues are created when OIM deploys:

- oimAttestationQueue
- oimAuditQueue
- oimDefaultQueue
- oimKernelQueue
- oimProcessQueue
- oimReconQueue
- oimSODQueue

The `xlconfig.xml` file stores Design Console and Remote Manager configuration.

External Dependencies

Oracle Identity Manager uses the Worklist and Human workflow modules of the Oracle SOA Suite for request flow management. OIM interacts with external repositories to store configuration and runtime data, and the repositories must be available during initialization and runtime. The OIM repository stores all OIM credentials. External components that OIM requires are:

- WebLogic Server
  - Administration Server
  - Managed Server
- Data Repositories
  - Configuration Repository (MDS Schema)
  - Runtime Repository (OIM Schema)
  - User Repository (OIM Schema)
  - SOA Repository (SOA Schema)
- BI Publisher, which can be optionally integrated with OIM

The Design Console is a tool used by the administrator for development and customization. The Design Console communicates directly with the OIM engine, so it relies on the same components that the OIM server relies on.

Remote Manager is an optional independent standalone application, which calls the custom APIs on the local system. It needs JAR files for custom APIs in its classpath.

Oracle Identity Governance Log File Locations

As a Java EE application deployed on WebLogic Server, all server log messages log to the server log file. OIM-specific messages log into the WebLogic Server diagnostic log file where the application is deployed.

WebLogic Server log files are in the directory:

DOMAIN_HOME/servers/*serverName*/logs

The three main log files are *serverName*.log, *serverName*.out, and *serverName*-diagnostic.log, where *serverName* is the name of the WebLogic Server. For example, if the WebLogic Server name is wls_OIM1, then the diagnostic log file name is wls_OIM1-diagnostic.log. Use Oracle Enterprise Manager Fusion Middleware Control to view log files.

Oracle Identity Governance High Availability Concepts

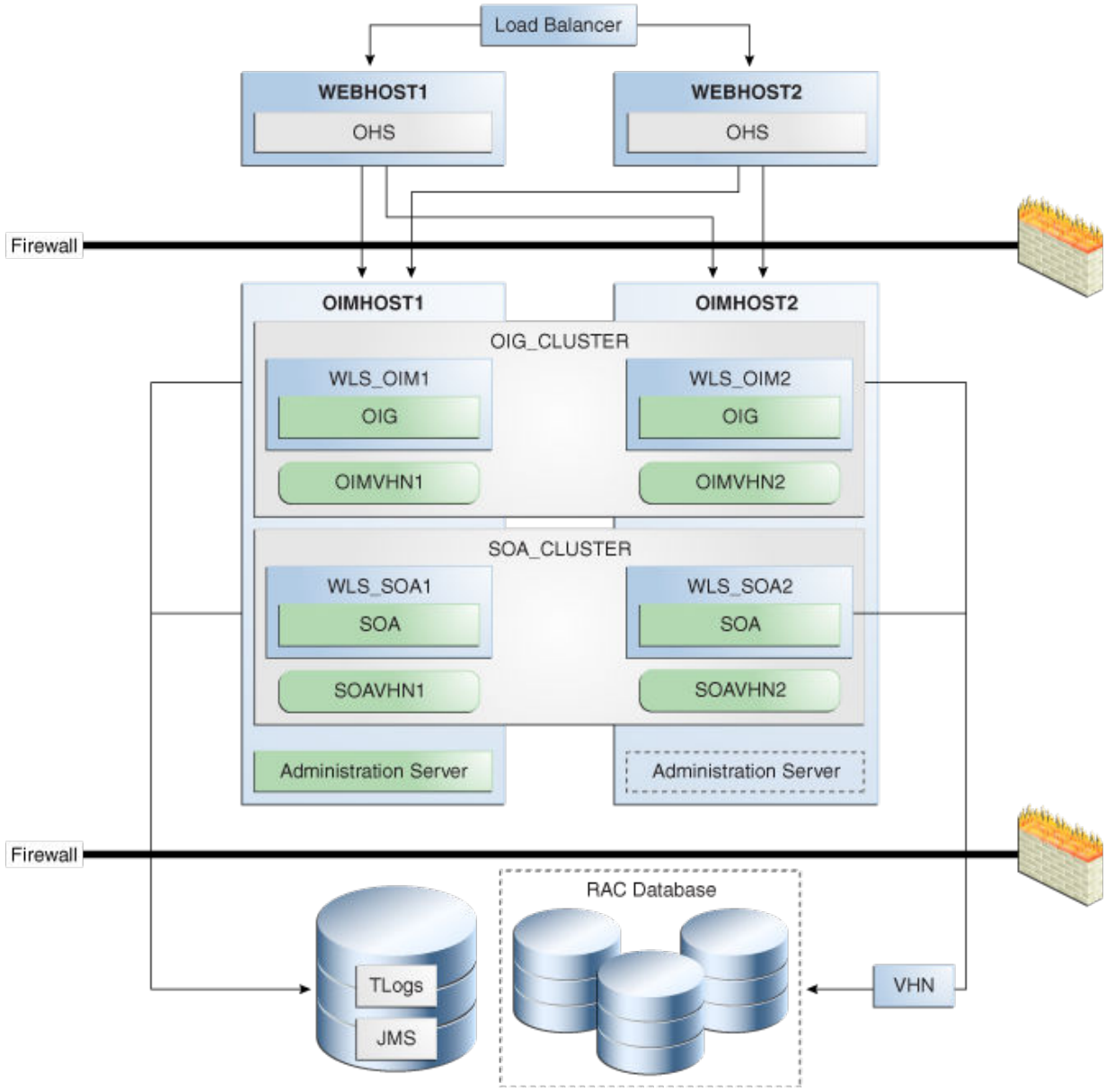
The concepts related to Oracle Identity Governance High Availability are OIG high availability architecture, starting and stopping OIG cluster, and cluster-wide configuration changes.

> **Note:**
>
> - You can deploy OIM on an Oracle RAC database, but Oracle RAC failover is not transparent for OIM in this release. If Oracle RAC failover occurs, end users may have to resubmit requests.
> - OIM always requires the availability of at least one node in the SOA cluster. If the SOA cluster is not available, end user requests fail. OIM does not retry for a failed SOA call. Therefore, the end user must retry when a SOA call fails.

Oracle Identity Governance High Availability Architecture

Figure 10-2 shows OIM deployed in a high availability architecture.

Figure 10-2 Oracle Identity Governance High Availability Architecture



On OIMHOST1, the following installations have been performed:

- An OIM instance is installed in the WLS_OIM1 Managed Server and a SOA instance is installed in the WLS_SOA1 Managed Server.
- The Oracle RAC database is configured in a GridLink data source to protect the instance from Oracle RAC node failure.
- A WebLogic Server Administration Server is been installed. Under normal operations, this is the active Administration Server.

On OIMHOST2, the following installations have been performed:

- An OIM instance is installed in the WLS_OIM2 Managed Server and a SOA instance is installed in the WLS_SOA2 Managed Server.

- The Oracle RAC database is configured in a GridLink data source to protect the instance from Oracle RAC node failure.
- The instances in the WLS_OIM1 and WLS_OIM2 Managed Servers on OIMHOST1 and OIMHOST2 are configured as the OIM_Cluster cluster.
- The instances in the WLS_SOA1 and WLS_SOA2 Managed Servers on OIMHOST1 and OIMHOST2 are configured as the SOA_Cluster cluster.
- An Administration Server is installed. Under normal operations, this is the passive Administration Server. You make this Administration Server active if the Administration Server on OIMHOST1 becomes unavailable.

Figure 10-2 uses these virtual host names in the OIM high availability configuration:

- OIMVHN1 is the virtual hostname that maps to the listen address for the WLS_OIM1 Managed Server, and it fails over with server migration of the WLS_OIM1 Managed Server. It is enabled on the node where the WLS_OIM1 Managed Server is running (OIMHOST1 by default).
- OIMVHN2 is the virtual hostname that maps to the listen address for the WLS_OIM2 Managed Server, and it fails over with server migration of the WLS_OIM2 Managed Server. It is enabled on the node where the WLS_OIM2 Managed Server is running (OIMHOST2 by default).
- SOAVHN1 is the virtual hostname that is the listen address for the WLS_SOA1 Managed Server, and it fails over with server migration of the WLS_SOA1 Managed Server. It is enabled on the node where the WLS_SOA1 Managed Server is running (OIMHOST1 by default).
- SOAVHN2 is the virtual hostname that is the listen address for the WLS_SOA2 Managed Server, and it fails over with server migration of the WLS_SOA2 Managed Server. It is enabled on the node where the WLS_SOA2 Managed Server is running (OIMHOST2 by default).
- VHN refers to the virtual IP addresses for the Oracle Real Application Clusters (Oracle RAC) database hosts.

Starting and Stopping the OIG Cluster

By default, WebLogic Server starts, stops, monitors, and manages lifecycle events for the application. The OIM application leverages high availability features of clusters. In case of hardware or other failures, session state is available to other cluster nodes that can resume the work of the failed node.

Use these command line tools and consoles to manage OIM lifecycle events:

- WebLogic Server Administration Console
- Oracle Enterprise Manager Fusion Middleware Control
- Oracle WebLogic Scripting Tool (WLST)

Cluster-Wide Configuration Changes

For high availability environments, changing the configuration of one OIM instance changes the configuration of all the other instances, because all the OIM instances share the same configuration repository.

High Availability Directory Structure Prerequisites

A high availability deployment requires product installations and files to reside in specific directories. A standard directory structure makes facilitates configuration across nodes and product integration.

Before you configure high availability, verify that your environment meets the requirements that High Availability Directory Structure Prerequisites describes.

Oracle Identity Governance High Availability Configuration Steps

Oracle Identity Governance high availability configuration involves setting the prerequisites, configuring the domain, post-installation steps, starting servers, SOA integration, validating managed server instances, and scaling up and scaling out Oracle Identity Governance.

This section provides high-level instructions for setting up a high availability deployment for OIM and includes these topics:

- Prerequisites for Configuring Oracle Identity Governance
- Configuring the Domain
- Post-Installation Steps on OIMHOST1
- Starting the Administration Server, oim_server1, and soa_server1
- Integrating Oracle Identity Governance with Oracle SOA Suite
- Propagating Oracle Identity Governance to OIMHOST2
- Post-Installation Steps on OIMHOST2
- Validate Managed Server Instances on OIMHOST2
- Configuring Server Migration for OIG and SOA Managed Servers
- Configuring a Default Persistence Store for Transaction Recovery
- Install Oracle HTTP Server on WEBHOST1 and WEBHOST2
- Configuring Oracle Identity Governance to Work with the Web Tier
- Validate the Oracle HTTP Server Configuration
- Oracle Identity Governance Failover and Expected Behavior
- Scaling Up Oracle Identity Governance
- Scaling Out Oracle Identity Governance

Prerequisites for Configuring Oracle Identity Governance

Before you configure OIM for high availability, you must:

- Install the Oracle Database. See Database Requirements in *Installing and Configuring Oracle Identity and Access Management*.

- Install the JDK on OIMHOST1 and OIMHOST2. See Preparing for Installation in *Installing and Configuring Oracle WebLogic Server and Coherence*.
- Install WebLogic Server, Oracle SOA Suite, and Oracle Identity Management software on OIMHOST1 and OIMHOST2 by using the quick installer. See Installing Oracle Identity Governance Using Quick Installer in *Installing and Configuring Oracle Identity and Access Management*.
- Run the Repository Creation Utility to create the OIM schemas in a database. See Running RCU to Create the OIM Schemas in a Database.
- Running RCU to Create the OIM Schemas in a Database

Running RCU to Create the OIM Schemas in a Database

The schemas you create depend on the products you want to install and configure. Use a Repository Creation Utility (RCU) that is version compatible with the product you install. See Creating the Database Schemas in *Installing and Configuring Oracle Identity and Access Management*.

Configuring the Domain

Use the Configuration Wizard to create and configure a domain.

See Configuring the Domain in *Installing and Configuring Oracle Identity and Access Management* for information about creating the Identity Management domain.

Post-Installation Steps on OIMHOST1

This section describes post-installation steps for OIMHOST1.

- Running the Offline Configuration Command
- Updating the System Properties for SSL Enabled Servers

Running the Offline Configuration Command

After you configure the Oracle Identity Governance domain, run the `offlineConfigManager` script to perform post configuration tasks.

Ensure that you run this command before you start any server. To run the `offlineConfigManager` command, do the following:

1. Set the following environment variables to the right values:
   - DOMAIN_HOME
   - JAVA_HOME
2. Ensure that you have execute permissions for the file *`OIM_HOME`*`/server/bin/offlineConfigManager.sh`.
3. Run the following command from the location *`OIM_HOME`*`/server/bin/`:
   - On Unix: `./offlineConfigManager.sh`
   - On Windows: `offlineConfigManager.bat`

Updating the System Properties for SSL Enabled Servers

For SSL enabled servers, you must set the required properties in the `setDomainEnv` file in the domain home.

Set the following properties in the *DOMAIN_HOME*`/bin/setDomainEnv.sh` (for UNIX) or *DOMAIN_HOME*`\bin\setDomainEnv.cmd` (for Windows) file before you start the servers:

- `-Dweblogic.security.SSL.ignoreHostnameVerification=true`
- `-Dweblogic.security.TrustKeyStore=DemoTrust`

Starting the Administration Server, oim_server1, and soa_server1

To start the Administration Server, oim_server1, and soa_server1:

1. To start the Administration Server, go to the DOMAIN_HOME/bin directory, and enter the following command:

   For UNIX: `./startWebLogic.sh`

   For Windows: `startWebLogic.cmd`

   If you selected Production Mode on the Domain Mode and JDK screen when you created the domain, you see a prompt for the Administrator user login credentials as the Administrator Account screen provides.
   You can verify that the Administration Server is up and running by accessing the Administration Server Console. The URL is provided on the End of Configuration screen (`http://administration_server_host:administration_server_port/console`). The default Administration Server port number is `7001`.

   > **Note:**
   >
   > Make sure that the database hosting your product schemas is up and running and accessible by the Administration Server.

2. Start the Node Manager on HOST1. To do so, run the following command from the *DOMAIN_HOME*/bin directory:

   (UNIX) Using `nohup` and `nm.out` as an example output file:

   `nohup ./startNodeManager.sh > LOG_DIR/nm.out&`

   Here, *LOG_DIR* is the location of directory in which you want to store the log files.

   (Windows) `startNodeManager.cmd`

3. Start the Oracle SOA Suite Managed Server(s) first and then the Oracle Identity Governance Managed Server(s). To start the Managed Servers:

   a. Login to Oracle Fusion Middleware Control:

      `http://administration_server_host:administration_server_port/em`

      The Enterprise Manager landing page lists servers configured for this domain and shows their status (such as Running or Shutdown). For a newly configured domain, only the **AdminServer(admin)** will be running.

b. Select wls_soa1.

c. From the Control list, select **Start**.

d. Repeat Steps b and c to start wls_oim1.

> **Note:**
>
> Ensure that you start the servers in the following order:
>
> i. Node Manager
>
> ii. Administration Server
>
> iii. Oracle SOA Suite Managed Server
>
> iv. Oracle Identity Manager Managed Server

e. On the main landing page, verify that all Managed Servers are up and running.

Integrating Oracle Identity Governance with Oracle SOA Suite

To integrate Oracle Identity Governance with Oracle SOA Suite:

1. Log in to Oracle Fusion Middleware Control by navigating to the following URL:

   http://administration_server_host:administration_server_port/em

2. Click **weblogic_domain**, and then select **System Mbean Browser**.

3. In the search box, enter `OIMSOAIntegrationMBean`, and click **Search**. The mbean is displayed.

> **Note:**
>
> If Oracle Identity Governance is still starting (coming up) or is just started (RUNNING MODE), then Enterprise Manager does not show any Mbeans defined by OIG. Wait for two minutes for the server to start, and then try searching for the Mbean in System Mbean Browser of the Enterprise Manager.

4. Click the **Operations** tab of mbean, and select **integrateWithSOAServer**.

5. Enter the required attributes, and then click **Invoke**.

Propagating Oracle Identity Governance to OIMHOST2

After the configuration succeeds on OIMHOST1, you can propagate it to OIMHOST2 by packing the domain on OIMHOST1 and unpacking it on OIMHOST2.

> **Note:**
>
> Oracle recommends that you perform a clean shut down of all Managed Servers on OIMHOST1 before you propagate the configuration to OIMHOST2.

To pack the domain on OIMHOST1 and unpack it on OIMHOST2:

1. On OIMHOST1, invoke the `pack` utility in the *ORACLE_HOME*`/oracle_common/common/bin` directory:

   ```
   pack.sh -domain=ORACLE_HOME/user_projects/domains/OIM_Domain -
   template =/u01/app/oracle/admin/templates/oim_domain.jar -
   template_name="OIM Domain" -managed=true
   ```

2. The previous step created the `oim_domain.jar` file in the following directory:

   ```
   /u01/app/oracle/admin/templates
   ```

   Copy `oim_domain.jar` from OIMHOST1 to a temporary directory on OIMHOST2.

3. On OIMHOST2, invoke the `unpack` utility in the *MW_HOME*`/oracle_common/common/bin` directory and specify the `oim_domain.jar` file location in its temporary directory:

   ```
   unpack.sh -domain=ORACLE_HOME/user_projects/domains/OIM_Domain -
   template=/tmp/oim_domain.jar
   ```

Post-Installation Steps on OIMHOST2

- Start Node Manager on OIMHOST2
- Start WLS_SOA2 and WLS_OIM2 Managed Servers on OIMHOST2

Start Node Manager on OIMHOST2

Start the Node Manager on OIMHOST2 using the `startNodeManager.sh` script located under the following directory:

DOMAIN_HOME`/bin`

Start WLS_SOA2 and WLS_OIM2 Managed Servers on OIMHOST2

To start Managed Servers on OIMHOST2:

1. Start the WLS_SOA2 Managed Server using the Administration Console.
2. Start the WLS_OIM2 Managed Server using the Administration Console. The WLS_OIM2 Managed Server must be started after the WLS_SOA2 Managed Server is started.

Validate Managed Server Instances on OIMHOST2

Validate the Oracle Identity Manager (OIM) and BI Publisher Managed Server instances on OIMHOST2.

Open the OIM Console with this URL:

```
http://identityvhn2.example.com:14000/oim
```

Log in using the `xelsysadm` password.

The URL for the BI Publisher is:

```
http://identityvhn2.example.com:9704/xmlpserver
```

Log in using the `xelsysadm` password.

Configuring Server Migration for OIG and SOA Managed Servers

For this high availability topology, Oracle recommends that you configure server migration for the WLS_OIM1, WLS_SOA1, WLS_OIM2, and WLS_SOA2 Managed Servers. See Section 3.9, "Whole Server Migration" for information on the benefits of using Whole Server Migration and why Oracle recommends it.

- The WLS_OIM1 and WLS_SOA1 Managed Servers on OIMHOST1 are configured to restart automatically on OIMHOST2 if a failure occurs on OIMHOST1.
- The WLS_OIM2 and WLS_SOA2 Managed Servers on OIMHOST2 are configured to restart automatically on OIMHOST1 if a failure occurs on OIMHOST2.

In this configuration, the WLS_OIM1, WLS_SOA1, WLS_OIM2 and WLS_SOA2 servers listen on specific floating IPs that WebLogic Server Migration fails over.

BI Publisher can be optionally integrated with OIG.

The subsequent topics enable server migration for the WLS_OIM1, WLS_SOA1, WLS_OIM2, and WLS_SOA2 Managed Servers, which in turn enables a Managed Server to fail over to another node if a server or process failure occurs.

- Editing Node Manager's Properties File
- Setting Environment and Superuser Privileges for the wlsifconfig.sh Script
- Configuring Server Migration Targets
- Testing the Server Migration

Editing Node Manager's Properties File

You must edit the `nodemanager.properties` file to add the following properties for each node where you configure server migration:

```
Interface=eth0
eth0=*,NetMask=255.255.248.0
UseMACBroadcast=true
```

- `Interface`: Specifies the interface name for the floating IP (such as `eth0`).

> **Note:**
>
> Do not specify the sub interface, such as `eth0:1` or `eth0:2`. This interface is to be used without the `:0`, or `:1`. The Node Manager's scripts traverse the different `:X` enabled IPs to determine which to add or remove. For example, valid values in Linux environments are `eth0`, `eth1`, or, `eth2`, `eth3`, `eth`*n*, depending on the number of interfaces configured.

- `NetMask`: Net mask for the interface for the floating IP. The net mask should be the same as the net mask on the interface; `255.255.255.0` is an example. The actual value depends on your network.
- `UseMACBroadcast`: Specifies whether or not to use a node's MAC address when sending ARP packets, that is, whether or not to use the `-b` flag in the `arping` command.

Verify in Node Manager's output (shell where Node Manager starts) that these properties are being used or problems may arise during migration. (Node Manager must be restarted to do this.) You should see an entry similar to the following in Node Manager's output:

```
...
StateCheckInterval=500
Interface=eth0
NetMask=255.255.255.0
...
```

Setting Environment and Superuser Privileges for the wlsifconfig.sh Script

To set environment and superuser privileges for the `wlsifconfig.sh` script for each node where you configure server migration:

1. Modify the login profile of the user account that you use to run Node Manager to ensure that the PATH environment variable for the Node Manager process includes directories housing the `wlsifconfig.sh` and `wlscontrol.sh` scripts, and the `nodemanager.domains` configuration file. Ensure that your PATH environment variable includes these files:

Table 10-1 Files Required for the PATH Environment Variable

File	Located in this directory
wlsifconfig.sh	*DOMAIN_HOME*/bin/server_migration
wlscontrol.sh	*WL_HOME*/common/bin
nodemanager.domains	*WL_HOME*/common

2. Grant sudo configuration for the `wlsifconfig.sh` script.
   - Configure sudo to work without a password prompt.
   - For security reasons, Oracle recommends restricting to the subset of commands required to run the `wlsifconfig.sh` script. For example, perform the following steps to set the environment and superuser privileges for the `wlsifconfig.sh` script:
   - Grant sudo privilege to the WebLogic user (`oracle`) with no password restriction, and grant execute privilege on the `/sbin/ifconfig` and `/sbin/arping binaries`.
   - Ensure that the script is executable by the WebLogic user. The following is an example of an entry inside `/etc/sudoers` granting sudo execution privilege for `oracle` and also over `ifconfig` and `arping`:

     ```
     oracle ALL=NOPASSWD: /sbin/ifconfig,/sbin/arping
     ```

> **Note:**
>
> Ask the system administrator for the sudo and system rights as appropriate to this step.

Configuring Server Migration Targets

You first assign all available nodes for the cluster's members and then specify candidate machines (in order of preference) for each server that is configured with server migration. To configure cluster migration in a cluster:

1. Log into the Administration Console.
2. In the Domain Structure window, expand **Environment** and select **Clusters**.
3. Click the cluster you want to configure migration for in the Name column.
4. Click the **Migration** tab.
5. Click **Lock and Edit**.
6. In the **Available** field, select the machine to which to enable migration and click the right arrow.
7. Select the data source to use for automatic migration. In this case, select the leasing data source, which is `WLSSchemaDataSource`.
8. Click **Save**.
9. Click **Activate Changes**.
10. Set the candidate machines for server migration. You must perform this task for all Managed Servers as follows:
    a. In the Domain Structure window of the Administration Console, expand **Environment** and select **Servers**.

    > **Tip:**
    >
    > Click **Customize this table** in the Summary of Servers page and move Current Machine from the Available window to the Chosen window to view the machine that the server runs on. This will be different from the configuration if the server migrates automatically.

    b. Select the server that you want to configure migration for.
    c. Click the **Migration** tab.
    d. In the **Available** field, located in the Migration Configuration section, select the machines you want to enable migration to and click the right arrow.
    e. Select **Automatic Server Migration Enabled**. This enables Node Manager to start a failed server on the target node automatically.
    f. Click **Save** then Click **Activate Changes**.
    g. Repeat the steps above for any additional Managed Servers.
    h. Restart the administration server, Node Managers, and the servers for which server migration has been configured.

Testing the Server Migration

To verify that server migration works properly:

From OIMHOST1:

1. Stop the WLS_OIM1 Managed Server by running the command:

   ```
   OIMHOST1> kill -9 pid
   ```

   where *pid* specifies the process ID of the Managed Server. You can identify the pid in the node by running this command:

   ```
   OIMHOST1> ps -ef | grep WLS_OIM1
   ```

2. Watch the Node Manager console. You should see a message indicating that WLS_OIM1's floating IP has been disabled.
3. Wait for Node Manager to try a second restart of WLS_OIM1. It waits for a fence period of 30 seconds before trying this restart.
4. Once Node Manager restarts the server, stop it again. Node Manager should now log a message indicating that the server will not be restarted again locally.

From OIMHOST2:

1. Watch the local Node Manager console. After 30 seconds since the last try to restart WLS_OIM1 on OIMHOST1, Node Manager on OIMHOST2 should prompt that the floating IP for WLS_OIM1 is being brought up and that the server is being restarted in this node.
2. Access the soa-infra console in the same IP.

Follow the steps above to test server migration for the WLS_OIM2, WLS_SOA1, and WLS_SOA2 Managed Servers.

Table 10-2 shows the Managed Servers and the hosts they migrate to in case of a failure.

Table 10-2 WLS_OIM1, WLS_OIM2, WLS_SOA1, WLS_SOA2 Server Migration

Managed Server	Migrated From	Migrated To
WLS_OIM1	OIMHOST1	OIMHOST2
WLS_OIM2	OIMHOST2	OIMHOST1
WLS_SOA1	OIMHOST1	OIMHOST2
WLS_SOA2	OIMHOST2	OIMHOST1

From Verification From the Administration Console

To verify migration in the Administration Console:

1. Log into the Administration Console at http://oimhost1.example.com:7001/console using administrator credentials.
2. Click **Domain** on the left console.
3. Click the **Monitoring** tab and then the **Migration** sub tab.

   The Migration Status table provides information on the status of the migration.

> **Note:**
>
> After a server migrates, to fail it back to its original node/machine, stop the Managed Server in the Administration Console then start it again. The appropriate Node Manager starts the Managed Server on the machine it was originally assigned to.

Configuring a Default Persistence Store for Transaction Recovery

Each Managed Server has a transaction log that stores information about in-flight transactions that the Managed Server coordinates that may not complete. WebLogic Server uses the transaction log to recover from system/network failures. To leverage the Transaction Recovery Service migration capability, store the transaction log in a location that all Managed Servers in a cluster can access. Without shared storage, other servers in the cluster can't run transaction recovery in the event of a server failure, so the operation may need to be retried.

> **Note:**
>
> Oracle recommends a location on a Network Attached Storage (NAS) device or Storage Area Network (SAN).

To set the location for default persistence stores for the OIM and SOA Servers:

1. Log into the Administration Console at http://oimhost1.*example*.com:7001/console using administrator credentials.
2. In the Domain Structure window, expand the **Environment** node and then click the **Servers** node. The Summary of Servers page opens.
3. Select the name of the server (represented as a hyperlink) in the **Name** column of the table. The Settings page for the server opens to the Configuration tab.
4. Select the **Services** subtab of the Configuration tab (not the Services top-level tab).
5. In the Default Store section, enter the path to the folder where the default persistent stores store their data files. The directory structure of the path should be:
   - For the WLS_SOA1 and WLS_SOA2 servers, use a directory structure similar to:

     `ORACLE_BASE/admin/domainName/soaClusterName/tlogs`
   - For the WLS_OIM1 and WLS_OIM2 servers, use a directory structure similar to:

     `ORACLE_BASE/admin/domainName/oimClusterName/tlogs`
6. Click **Save**.

> **Note:**
>
> To enable migration of Transaction Recovery Service, specify a location on a persistent storage solution that is available to the Managed Servers in the cluster. WLS_SOA1, WLS_SOA2, WLS_OIM1, and WLS_OIM2 must be able to access this directory.

Install Oracle HTTP Server on WEBHOST1 and WEBHOST2

Install Oracle HTTP Server on WEBHOST1 and WEBHOST2.

Configuring Oracle Identity Governance to Work with the Web Tier

The following topics describe how to configure OIM to work with the Oracle Web Tier.

- Prerequisites to Configure OIG to Work with the Web Tier
- Configuring Oracle HTTP Servers to Front End OIM, and SOA Managed Servers

Prerequisites to Configure OIG to Work with the Web Tier

Verify that the following tasks have been performed:

1. Oracle Web Tier has been installed on WEBHOST1 and WEBHOST2.
2. OIM is installed and configured on OIMHOST1 and OIMHOST2.
3. The load balancer has been configured with a virtual hostname (`sso.example.com`) pointing to the web servers on WEBHOST1 and WEBHOST2. I`sso.example.com`is customer facing and the main point of entry; it is typically SSL terminated.
4. The load balancer has been configured with a virtual hostname (`oiminternal.example.com`) pointing to web servers WEBHOST1 and WEBHOST2. `oiminternal.example.com` is for internal callbacks and is *not* customer facing.

Configuring Oracle HTTP Servers to Front End OIM, and SOA Managed Servers

bug 18547492 for last code example in step #1

1. On each of the web servers on `WEBHOST1` and `WEBHOST2`, create a file named `oim.conf` in the directory *`ORACLE_INSTANCE`*`/config/OHS/`*`COMPONENT`*`/moduleconf`.

   This file must contain the following information:

```
# oim admin console(idmshell based)
   <Location /admin>
    SetHandler weblogic-handler
    WLCookieName    oimjsessionid
    WebLogicCluster oimvhn1.example.com:14000,oimvhn2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    WLProxySSL ON
    WLProxySSLPassThrough ON
```

```
   </Location>

# oim self and advanced admin webapp consoles(canonic webapp)

  <Location /oim>
    SetHandler weblogic-handler
    WLCookieName    oimjsessionid
    WebLogicCluster oimvhn1.example.com:14000,oimvhn2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    WLProxySSL ON
    WLProxySSLPassThrough ON
   </Location>

  <Location /identity>
    SetHandler weblogic-handler
    WLCookieName    oimjsessionid
    WebLogicCluster oimvhn1.example.com:14000,oimvhn2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    WLProxySSL ON
    WLProxySSLPassThrough ON
    </Location>

  <Location /sysadmin>
    SetHandler weblogic-handler
    WLCookieName    oimjsessionid
    WebLogicCluster oimvhn1.example.com:14000,oimvhn2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    WLProxySSL ON
    WLProxySSLPassThrough ON
    </Location>

# SOA Callback webservice for SOD - Provide the SOA Managed Server Ports
  <Location /sodcheck>
    SetHandler weblogic-handler
    WLCookieName    oimjsessionid
    WebLogicCluster soavhn1.example.com:7003,soavhn2.example.com:7003
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    WLProxySSL ON
    WLProxySSLPassThrough ON
   </Location>

# Callback webservice for SOA. SOA calls this when a request is approved/rejected
# Provide the OIM Managed Server Port
  <Location /workflowservice>
    SetHandler weblogic-handler
    WLCookieName    oimjsessionid
    WebLogicCluster oimvhn1.example.com:14000,oimvhn2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    WLProxySSL ON
    WLProxySSLPassThrough ON
  </Location>

# xlWebApp - Legacy 9.x webapp (struts based)
   <Location /xlWebApp>
    SetHandler weblogic-handler
    WLCookieName    oimjsessionid
    WebLogicCluster oimvhn1.example.com:14000,oimvhn2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    WLProxySSL ON
    WLProxySSLPassThrough ON
  </Location>
```

```
# Nexaweb WebApp - used for workflow designer and DM
  <Location /Nexaweb>
    SetHandler weblogic-handler
    WLCookieName    oimjsessionid
    WebLogicCluster oimvhn1.example.com:14000,oimvhn2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    WLProxySSL ON
    WLProxySSLPassThrough ON
  </Location>

# used for FA Callback service.
  <Location /callbackResponseService>
    SetHandler weblogic-handler
    WLCookieName    oimjsessionid
    WebLogicCluster oimvhn1.example.com:14000,oimvhn2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    WLProxySSL ON
    WLProxySSLPassThrough ON
  </Location>

# spml xsd profile
  <Location /spml-xsd>
    SetHandler weblogic-handler
    WLCookieName    oimjsessionid
    WebLogicCluster oimvhn1.example.com:14000,oimvhn2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    WLProxySSL ON
    WLProxySSLPassThrough ON
  </Location>

  <Location /HTTPClnt>
    SetHandler weblogic-handler
    WLCookieName    oimjsessionid
    WebLogicCluster oimvhn1.example.com:14000,oimvhn2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    WLProxySSL ON
    WLProxySSLPassThrough ON
  </Location>


  <Location /reqsvc>
    SetHandler weblogic-handler
    WLCookieName oimjsessionid
    WebLogicCluster oimvhn1.example.com:14000,oimvhn2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    WLProxySSL ON
    WLProxySSLPassThrough ON
  </Location>


  <Location /integration>
    SetHandler weblogic-handler
    WLCookieName oimjsessionid
    WebLogicCluster soavhn1.example.com:7003,soavhn2.example.com:7003
    WLProxySSL ON
    WLProxySSLPassThrough ON
  </Location>


  <Location /provisioning-callback>
```

```
   SetHandler weblogic-handler
   WLCookieName oimjsessionid
   WebLogicCluster oimvhn1.example.com:14000,oimvhn2.example.com:14000
   WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
   WLProxySSL ON
   WLProxySSLPassThrough ON
 </Location>

 <Location /CertificationCallbackService>
  SetHandler weblogic-handler
  WLCookieName JSESSIONID
  WebLogicCluster oimvhn1.example.com:14000,oimvhn2.example.com:14000
  WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
  WLProxySSL ON
  WLProxySSLPassThrough ON
</Location>

 <Location /ucs>
  SetHandler weblogic-handler
  WLCookieName oimjsessionid
  WebLogicCluster soavhn1.example.com:7003,soavhn2.example.com:7003
  WLLogFile /tmp/web_log.log
  WLProxySSL ON
  WLProxySSLPassThrough ON
 </Location>

 <Location /FacadeWebApp>
  SetHandler weblogic-handler
  WLCookieName oimjsessionid
  WebLogicCluster oimvhn1.example.com:14000,oimvhn2.example.com:14000
  WLLogFile /tmp/web_log.log
  WLProxySSL ON
  WLProxySSLPassThrough ON
 </Location>

 <Location /iam/governance/configmgmt>
  SetHandler weblogic-handler
  WLCookieName oimjsessionid
  WebLogicCluster oimvhn1.example.com:14000,oimvhn2.example.com:14000
  WLLogFile /tmp/web_log.log
  WLProxySSL ON
  WLProxySSLPassThrough ON
 </Location>

 <Location /iam/governance/scim/v1>
  SetHandler weblogic-handler
  WLCookieName oimjsessionid
  WebLogicCluster oimvhn1.example.com:14000,oimvhn2.example.com:14000
  WLLogFile /tmp/web_log.log
  WLProxySSL ON
  WLProxySSLPassThrough ON>
 </Location>

 <Location /iam/governance/token/api/v1>
  SetHandler weblogic-handler
  WLCookieName oimjsessionid
  WebLogicCluster oimvhn1.example.com:14000,oimvhn2.example.com:14000
  WLLogFile /tmp/web_log.log
  WLProxySSL ON
  WLProxySSLPassThrough ON
 </Location>
```

```
<Location /OIGUI>
 SetHandler weblogic-handler
 WLCookieName oimjsessionid
 WebLogicCluster oimvhn1.example.com:14000,oimvhn2.example.com:14000
 WLLogFile /tmp/web_log.log
 WLProxySSL ON
 WLProxySSLPassThrough ON
</Location>

<Location /iam/governance/applicationmanagement>
 SetHandler weblogic-handler
 WLCookieName oimjsessionid
 WebLogicCluster oimvhn1.example.com:14000,oimvhn2.example.com:14000
 WLLogFile /tmp/web_log.log
 WLProxySSL ON
 WLProxySSLPassThrough ON
</Location>

<Location /iam/governance/adminservice/api/v1>
 SetHandler weblogic-handler
 WLCookieName oimjsessionid
 WebLogicCluster oimvhn1.example.com:14000,oimvhn2.example.com:14000
 WLLogFile /tmp/web_log.log
 WLProxySSL ON
 WLProxySSLPassThrough ON
</Location>

<Location /iam/governance/selfservice/api/v1>
 SetHandler weblogic-handler
 WLCookieName oimjsessionid
 WebLogicCluster oimvhn1.example.com:14000,oimvhn2.example.com:14000
 WLLogFile /tmp/web_log.log
 WLProxySSL ON
 WLProxySSLPassThrough ON
</Location>
```

2. Create a file called `virtual_hosts.conf` in *ORACLE_INSTANCE*`/config/OHS/`*COMPONENT*`/moduleconf`. The file must contain the following information:

> **Note:**
>
> *COMPONENT* is typically `ohs1` or `ohs2`. However, the name depends on choices you made during OHS installation.

```
NameVirtualHost *:7777
<VirtualHost *:7777>

  ServerName http://sso.example.com:7777
  RewriteEngine On
  RewriteOptions inherit
  UseCanonicalName On
  </VirtualHost>

<VirtualHost *:7777>
  ServerName http://oiminternal.example.com:80
  RewriteEngine On
  RewriteOptions inherit
```

```
  UseCanonicalName On
</VirtualHost>
```

3. Save the file on both `WEBHOST1` and `WEBHOST2`.
4. Stop and start the Oracle HTTP Server instances on both `WEBHOST1` and `WEBHOST2`.

Validate the Oracle HTTP Server Configuration

To validate that Oracle HTTP Server is configured properly, follow these steps:

1. In a web browser, enter the following URL for the Oracle Identity Manager Console:

   ```
   http://sso.example.com:7777/identity
   ```

   The Oracle Identity Manager Console login page should display.
2. Log into the Oracle Identity Manager Console using the credentials for the `xelsysadm` user.

Oracle Identity Governance Failover and Expected Behavior

In a high availability environment, you configure Node Manager to monitor Oracle WebLogic Servers. In case of failure, Node Manager restarts the WebLogic Server.

A hardware load balancer load balances requests between multiple OIM instances. If one OIM Managed Server fails, the load balancer detects the failure and routes requests to surviving instances.

In a high availability environment, state and configuration information is stored in a database that all cluster members share. Surviving OIM instances continue to seamlessly process any unfinished transactions started on the failed instance because state information is in the shared database, available to all cluster members.

When an OIM instance fails, its database and LDAP connections are released. Surviving instances in the active-active deployment make their own connections to continue processing unfinished transactions on the failed instance.

When you deploy OIM in a high availability configuration:

- You can deploy OIM on an Oracle RAC database, but Oracle RAC failover is not transparent for OIM in this release. If Oracle RAC failover occurs, end users may have to resubmit their requests.
- Oracle Identity Manager always requires the availability of at least one node in the SOA cluster. If the SOA cluster is not available, end user requests fail. OIM does not retry for a failed SOA call. Therefore, the end user must retry when a SOA call fails.

Scaling Up Oracle Identity Governance

You can scale out or scale up the OIG high availability topology. When you *scale up* the topology, you add new Managed Servers to nodes that are already running one or more Managed Servers. When you *scale out* the topology, you add new Managed Servers to new nodes. See Scaling Out Oracle Identity Governance to scale out.

In this case, you have a node that runs a Managed Server configured with SOA. The node contains:

- A Middleware home

- An Oracle HOME (SOA)
- A domain directory for existing Managed Servers

You can use the existing installations (Middleware home and domain directories) to create new WLS_OIM and WLS_SOA Managed Servers. You do not need to install OIM and SOA binaries in a new location or run pack and unpack.

This procedure describes how to clone OIM and SOA Managed Servers. You may clone one or two of these component types, as long as one of them is OIM.

Note the following:

- This procedure refers to WLS_OIM and WLS_SOA. However, you may not be scaling up both the components. For each step, choose the component(s) that you are scaling up in your environment. Also, some steps do not apply to all components
- The persistent store's shared storage directory for JMS Servers must exist before you start the Managed Server or the start operation fails.
- Each time you specify the persistent store's path, it must be a directory on shared storage

To scale up the topology:

1. In the Administration Console, clone WLS_OIM1/WLS_SOA1. The Managed Server that you clone should be one that already exists on the node where you want to run the new Managed Server.
   a. Select **Environment -> Servers** from the Administration Console.
   b. Select the Managed Server(s) that you want to clone.
   c. Select **Clone**.
   d. Name the new Managed Server WLS_OIM`n`/WLS_SOA`n`, where `n` is a number to identity the new Managed Server.

   The rest of the steps assume that you are adding a new Managed Server to OIMHOST1, which is already running WLS_OIM1 and WLS_SOA1.
2. For the listen address, assign the hostname or IP for the new Managed Server(s). If you plan to use server migration, use the VIP (floating IP) to enable Managed Server(s) to move to another node. Use a VIP different from the VIP that the existing Managed Server uses.
3. Create JMS Servers for OIM/SOA, BPM, UMS, JRFWSAsync, and SOAJMServer on the new Managed Server.
   a. In the Administration Console, create a new persistent store for the OIM JMS Server(s) and name it. Specify the store's path, a directory on shared storage.

      ORACLE_BASE/admin/*domain_name*/*cluster_name*/jms
   b. Create a new JMS Server for OIM. Use `JMSFileStore_`*n* for JMSServer. Target `JMSServer_`*n* to the new Managed Server(s).
   c. Create a persistence store for the new UMSJMSServer(s), for example, `UMSJMSFileStore_`*n*. Specify the store's path, a directory on shared storage.

      ORACLE_BASE/admin/*domain_name*/*cluster_name*/jms/UMSJMSFileStore_n
   d. Create a new JMS Server for UMS, for example, `UMSJMSServer_`*n*. Target it to the new Managed Server (WLS_SOAn).

e. Create a persistence store for the new BPMJMSServer(s), for example, `BPMJMSFileStore_n`. Specify the store's path, a directory on shared storage.

   ORACLE_BASE/admin/*domain_name*/*cluster_name*/jms/BPMJMSFileStore_n

f. Create a new JMS Server for BPM, for example, `BPMJMSServer_n`. Target it to the new Managed Server (WLS_SOAn).

g. Create a new persistence store for the new JRFWSAsyncJMSServer, for example, `JRFWSAsyncJMSFileStore_n`. Specify the store's path, a directory on shared storage.

   ORACLE_BASE/admin/*domain_name*/*cluster_name*/jms/JRFWSAsyncJMSFileStore_n

h. Create a JMS Server for JRFWSAsync, for example, `JRFWSAsyncJMSServer_n`. Use `JRFWSAsyncJMSFileStore_n` for this JMSServer. Target `JRFWSAsyncJMSServer_n` to the new Managed Server (WLS_OIMn).

> **Note:**
>
> You can also assign `SOAJMSFileStore_n` as store for the new JRFWSAsync JMS Servers. For clarity and isolation, individual persistent stores are used in the following steps.

i. Create a persistence store for the new SOAJMSServer, for example, `SOAJMSFileStore_auto_n`. Specify the store's path, a directory on shared storage.

   ORACLE_BASE/admin/*domain_name*/*cluster_name*/jms/SOAJMSFileStore_auto_n

j. Create a JMS Server for SOA, for example, `SOAJMSServer_auto_n`. Use `SOAJMSFileStore_auto_n` for this JMSServer. Target `SOAJMSServer_auto_n` to the new Managed Server (WLS_SOAn).

> **Note:**
>
> You can also assign `SOAJMSFileStore_n` as store for the new PS6 JMS Servers. For the purpose of clarity and isolation, individual persistent stores are used in the following steps.

k. Update SubDeployment targets for SOA JMS Module to include the new SOA JMS Server. Expand the **Services** node, then expand the **Messaging** node. Choose **JMS Modules** from the Domain Structure window. Click **SOAJMSModule** (hyperlink in the **Names** column). In the Settings page, click the **SubDeployments** tab. In the subdeployment module, click the **SOAJMSServerXXXXXX** subdeployment and add `SOAJMSServer_n` to it Click **Save**.

> **Note:**
>
> A subdeployment module name is a random name in the form *COMPONENT*JMSServerXXXXXX. It comes from the Configuration Wizard JMS configuration for the first two Managed Servers, WLS_*COMPONENT*1 and WLS_*COMPONENT*2).

l. Update SubDeployment targets for UMSJMSSystemResource to include the new UMS JMS Server. Expand the **Services** node, then expand the **Messaging** node. Choose **JMS Modules** from the Domain Structure window. Click **UMSJMSSystemResource** (hyperlink in the Names column). In the Settings page, click the SubDeployments tab. In the subdeployment module, click the **UMSJMSServerXXXXXX** subdeployment and add `UMSJMSServer_`*n* to it. Click **Save**.

m. Update SubDeployment targets for OIMJMSModule to include the new OIM JMS Server. Expand the **Services** node, then expand **Messaging** node. Choose **JMS Modules** from the Domain Structure window. Click **OIMJMSModule** (hyperlink in the **Names** column). In the Settings page, click the SubDeployments tab. In the subdeployment module, click **OIMJMSServerXXXXXX** and `OIMJMSServer_n` to it. Click **Save**.

n. Update SubDeployment targets for the JRFWSAsyncJmsModule to include the new JRFWSAsync JMS Server. Expand the **Services** node then expand the **Messaging** node. Choose **JMS Modules** from the Domain Structure window. Click **JRFWSAsyncJmsModule** (hyperlink in the Names column of the table). In the Settings page, click the SubDeployments tab. Click the **JRFWSAsyncJMSServerXXXXXX** subdeployment and add `JRFWSAsyncJMSServer_`*n* to this subdeployment. Click **Save**

o. Update SubDeployment targets for BPM JMS Module to include the new BPM JMS Server. Expand the **Services** node, then expand the **Messaging** node. Choose **JMS Modules** from the Domain Structure window. Click **BPMJMSModule** (hyperlink in the **Names** column). In the Settings page, click the **SubDeployments** tab. In the subdeployment module, click the **BPMJMSServerXXXXXX** subdeployment and add `BPMJMSServer_`*n* to it. Click **Save**.

4. Configure the transaction persistent store for the new server in a shared storage location visible from other nodes.

   From the Administration Console, select **Server_name > Services** tab. Under Default Store, in Directory, enter the path to the default persistent store.

5. Disable hostname verification for the new Managed Server (required before starting/verifying a WLS_SOA*n* Managed Server) You can re-enable it after you configure server certificates for Administration Server / Node Manager communication in SOAHOSTn. If the source server (from which you cloned the new Managed Server) had disabled hostname verification, these steps are not required; hostname verification settings propagate to a cloned server.

   To disable hostname verification:

   a. In the Administration Console, expand the **Environment** node in the Domain Structure window.

   b. Click **Servers**. Select WLS_SOAn in the **Names** column of the table.

   c. Click the SSL tab. Click **Advanced.**

   d. Set **Hostname Verification** to **None**. Click **Save**.

6. Start and test the new Managed Server from the Administration Console.

   a. Shut down the existing Managed Servers in the cluster.

   b. Ensure that the newly created Managed Server is up.

c. Access the application on the newly created Managed Server to verify that it works. A login page opens for OIM. For SOA, a HTTP basic authorization opens.

Table 10-3 Managed Server Test URLs

Component	Managed Server Test URL
SOA	http://*vip*:*port*/soa-infra
OIM	http://*vip*:*port*/identity

7. In the Administration Console, select **Services** then **Foreign JNDI provider**. Confirm that **ForeignJNDIProvider-SOA** targets `cluster:t3://soa_cluster`, not a Managed Server(s). You target the cluster so that new Managed Servers don't require configuration. If **ForeignJNDIProvider-SOA** does not target the cluster, target it to the cluster.
8. Configure Server Migration for the new Managed Server.

> **Note:**
>
> For scale up, the node must have a Node Manager, an environment configured for server migration, and the floating IP for the new Managed Server(s).

To configure server migration:

a. Log into the Administration Console.
b. In the left pane, expand **Environment** and select **Servers**.
c. Select the server (hyperlink) that you want to configure migration for.
d. Click the Migration tab.
e. In the Available field, in the Migration Configuration section, select machines to enable migration for and click the right arrow. Select the same migration targets as for the servers that already exist on the node.

   For example:

   For new Managed Servers on SOAHOST1, which is already running WLS_SOA1, select SOAHOST2.

   For new Managed Servers on SOAHOST2, which is already running WLS_SOA2, select SOAHOST1.

   Verify that the appropriate resources are available to run Managed Servers concurrently during migration.
f. Select the **Automatic Server Migration Enabled** option to enable Node Manager to start a failed server on the target node automatically.
g. Click **Save**.
h. Restart the Administration Server, Managed Servers, and Node Manager.
i. Repeat these steps to configure server migration for the newly created WLS_OIMn Managed Server.

9. To test server migration for this new server, follow these steps from the node where you added the new server:

a. Stop the Managed Server.

   Run `kill -9` *pid* on the PID of the Managed Server. To identify the PID of the node, enter, for example, `ps -ef | grep WLS_SOA`*n*.

b. Watch Node Manager Console for a message indicating that the Managed Server floating IP is disabled.

c. Wait for Node Manager to try a second restart of the Managed Server. Node Manager waits for 30 seconds before trying this restart.

d. After Node Manager restarts the server, stop it again. Node Manager logs a message indicating that the server will not restart again locally.

10. Edit the OHS configuration file to add the new managed server(s). See Configuring Oracle HTTP Server to Recognize New Managed Servers.

Scaling Out Oracle Identity Governance

When you scale out the topology, you add new Managed Servers configured with software to new nodes.

> **Note:**
>
> Steps in this procedure refer to WLS_OIM and WLS_SOA. However, you may not be scaling up both the components. For each step, choose the component(s) that you are scaling up in your environment. Some steps do not apply to all components.

Before you scale out, check that you meet these requirements:

- Existing nodes running Managed Servers configured with OIM and SOA in the topology.
- The new node can access existing home directories for WebLogic Server, SOA, and OIM. (Use existing installations in shared storage to create new Managed Server. You do not need to install WebLogic Server or component binaries in a new location, but must run pack and unpack to bootstrap the domain configuration in the new node.)

  > **Note:**
  >
  > If there is no existing installation in shared storage, you must install WebLogic Server, SOA, and OIM in the new nodes.

> **Note:**
>
> When multiple servers in different nodes share *ORACLE_HOME* or *WL_HOME*, Oracle recommends keeping the Oracle Inventory and Middleware home list in those nodes updated for consistency in the installations and application of patches. To update the oraInventory in a node and "attach" an installation in a shared storage to it, use *ORACLE_HOME*`/oui/bin/attachHome.sh`.

To scale out the topology:

1. On the new node, mount the existing Middleware home. Include the SOA and OIM installations and the domain directory, and ensure the new node has access to this directory, just like the rest of the nodes in the domain.
2. Attach *ORACLE_HOME* in shared storage to the local Oracle Inventory. For example:

   ```
   cd /u01/app/oracle/soa/
   ./attachHome.sh -jreLoc u01/app/JRE-JDK_version>
   ```

   To update the Middleware home list, create (or edit, if another WebLogic installation exists in the node) the *MW_HOME*`/bea/beahomelist` file and add `u01/app/oracle` to it.
3. Log in to the Administration Console.
4. Create a new machine for the new node. Add the machine to the domain.
5. Update the machine's Node Manager's address to map the IP of the node that is being used for scale out.
6. Clone WLS_OIM1/WLS_SOA1.

   To clone OIM and SOA:

   a. Select **Environment -> Servers** from the Administration Console.
   b. Select the Managed Server(s) that you want to clone.
   c. Select **Clone**.
   d. Name the new Managed Server WLS_OIM`n`/WLS_SOA`n`, where `n` is a number to identity the new Managed Server.

> **Note:**
>
> These steps assume that you are adding a new server to node *n*, where no Managed Server was running previously.

7. Assign the hostname or IP to use for the new Managed Server for the listen address of the Managed Server.

   If you plan to use server migration for this server (which Oracle recommends), this should be the server VIP (*floating IP*). This VIP should be different from the one used for the existing Managed Server.
8. Create JMS servers for SOA, OIM (if applicable), UMS, BPM, JRFWSAsync, and SOA on the new Managed Server.

   a. In the Administration Console, create a new persistent store for the OIM JMS Server and rename it. Specify the store's path, a directory on shared storage.

ORACLE_BASE/admin/*domain_name*/*cluster_name*/jms

b. Create a new JMS Server for OIM. Use JMSFileStore_*n* for JMSServer. Target JMSServer_*n* to the new Managed Server(s).

c. Create a persistence store for the new UMSJMSServer(s), for example, UMSJMSFileStore_*n*. Specify the store's path, a directory on shared storage.

ORACLE_BASE/admin/*domain_name*/*cluster_name*/jms/UMSJMSFileStore_n

d. Create a new JMS Server for UMS, for example, UMSJMSServer_*n*. Target it to the new Managed Server (WLS_SOAn).

e. Create a persistence store for the new BPMJMSServer(s), for example, BPMJMSFileStore_*n*. Specify the store's path, a directory on shared storage.

ORACLE_BASE/admin/*domain_name*/*cluster_name*/jms/BPMJMSFileStore_*n*

f. Create a new JMS Server for BPM, for example, BPMJMSServer_*n*. Target it to the new Managed Server (WLS_SOAn).

g. Create a new persistence store for the new JRFWSAsyncJMSServer, for example, JRFWSAsyncJMSFileStore_*n*. Specify the store's path, a directory on shared storage.

ORACLE_BASE/admin/*domain_name*/*cluster_name*/jms/JRFWSAsyncJMSFileStore_*n*

h. Create a JMS Server for JRFWSAsync, for example, JRFWSAsyncJMSServer_*n*. Use JRFWSAsyncJMSFileStore_*n* for this JMSServer. Target JRFWSAsyncJMSServer_*n* to the new Managed Server (WLS_OIMn).

> **Note:**
>
> You can also assign SOAJMSFileStore_*n* as store for the new JRFWSAsync JMS Servers. For clarity and isolation, the following steps use individual persistent stores.

i. Create a persistence store for the new SOAJMSServer, for example, SOAJMSFileStore_auto_*n*. Specify the store's path, a directory on shared storage.

ORACLE_BASE/admin/*domain_name*/*cluster_name*/jms/SOAJMSFileStore_auto_*n*

j. Create a JMS Server for SOA, for example, SOAJMSServer_auto_*n*. Use SOAJMSFileStore_auto_*n* for this JMSServer. Target SOAJMSServer_auto_*n* to the new Managed Server (WLS_SOAn).

> **Note:**
>
> You can also assign SOAJMSFileStore_*n* as store for the new PS6 JMS Servers. For clarity and isolation, the following steps use individual persistent stores.

k. Update SubDeployment targets for SOA JMS Module to include the new SOA JMS Server. Expand the **Services** node, then expand the **Messaging** node. Choose **JMS Modules** from the Domain Structure window. Click **SOAJMSModule** (hyperlink in the **Names** column). In the Settings page, click

the **SubDeployments** tab. In the subdeployment module, click the **SOAJMSServerXXXXXX** subdeployment and add `SOAJMSServer_n` to it Click **Save**.

> **Note:**
>
> A subdeployment module name is a random name in the form *COMPONENT*JMSServerXXXXXX. It comes from the Configuration Wizard JMS configuration for the first two Managed Servers, WLS_*COMPONENT*1 and WLS_*COMPONENT*2).

l. Update SubDeployment targets for UMSJMSSystemResource to include the new UMS JMS Server. Expand the **Services** node, then expand the **Messaging** node. Choose **JMS Modules** from the Domain Structure window. Click **UMSJMSSystemResource** (hyperlink in the Names column). In the Settings page, click the SubDeployments tab. In the subdeployment module, click the **UMSJMSServerXXXXXX** subdeployment and add `UMSJMSServer_n` to it. Click **Save**.

m. Update SubDeployment targets for OIMJMSModule to include the new OIM JMS Server. Expand the **Services** node, then expand **Messaging** node. Choose **JMS Modules** from the Domain Structure window. Click **OIMJMSModule** (hyperlink in the **Names** column). In the Settings page, click the SubDeployments tab. In the subdeployment module, click **OIMJMSServerXXXXXX** and `OIMJMSServer_n` to it. Click **Save**.

n. Update SubDeployment targets for the JRFWSAsyncJmsModule to include the new JRFWSAsync JMS Server. Expand the **Services** node then expand the **Messaging** node. Choose **JMS Modules** from the Domain Structure window. Click **JRFWSAsyncJmsModule** (hyperlink in the Names column). In the Settings page, click the SubDeployments tab. Click the **JRFWSAsyncJMSServerXXXXXX** subdeployment and add `JRFWSAsyncJMSServer_n` to this subdeployment. Click **Save**

o. Update SubDeployment targets for BPM JMS Module to include the new BPM JMS Server. Expand the **Services** node, then expand the **Messaging** node. Choose **JMS Modules** from the Domain Structure window. Click **BPMJMSModule** (hyperlink in the **Names** column). In the Settings page, click the **SubDeployments** tab. In the subdeployment module, click the **BPMJMSServerXXXXXX** subdeployment and add `BPMJMSServer_n` to it. Click **Save**.

9. Run the pack command on SOAHOST1 to create a template pack. For example:

```
cd ORACLE_HOME/oracle_common/common/bin
./pack.sh -managed=true/
-domain=MW_HOME/user_projects/domains/soadomain/
-template=soadomaintemplateScale.jar -template_name='soa_domain_templateScale'
```

Run the following command on HOST1 to copy the template file created to HOST*n*:

```
scp soadomaintemplateScale.jar oracle@SOAHOSTN:/
ORACLE_BASE/product/fmw/soa/common/bin
```

Run the unpack command on HOSTn to unpack the template in the Managed Server domain directory. For example, for SOA:

```
ORACLE_HOME/oracle_common/common/bin
/unpack.sh /
-domain=ORACLE_BASE/product/fmw/user_projects/domains/soadomain/
-template=soadomaintemplateScale.jar
```

10. Configure the transaction persistent store for the new server. This should be a shared storage location visible from other nodes.

    From the Administration Console, select **Server_name > Services** tab. Under Default Store, in Directory, enter the path to the folder where you want the default persistent store to store its data files.

11. Disable hostname verification for the new Managed Server; you must do this before starting/verifying the Managed Server. You can re-enable it after you configure server certificates for the communication between the Administration Server and Node Manager. If the source Managed Server (server you cloned the new one from) had already disabled hostname verification, these steps are not required. Hostname verification settings propagate to cloned servers.

    To disable hostname verification:

    a. Open the Administration Console.

    b. Expand the **Environment** node in the Domain Structure window.

    c. Click **Servers**.

    d. Select WLS_SOAn in the **Names** column of the table. The Settings page for the server appears.

    e. Click the SSL tab.

    f. Click **Advanced**.

    g. Set **Hostname Verification** to **None**.

    h. Click **Save**.

12. Start Node Manager on the new node. To start the Node Manager, use the installation in shared storage from the existing nodes, and start Node Manager by passing the hostname of the new node as a parameter as follows:

    ```
    WL_HOME/server/bin/startNodeManager new_node_ip
    ```

13. Start and test the new Managed Server from the Administration Console.

    a. Shut down the existing Managed Server in the cluster.

    b. Ensure that the newly created Managed Server is up.

    c. Access the application on the newly created Managed Server to verify that it works. A login page appears for OIM. For SOA, a HTTP basic authorization opens.

Table 10-4 Managed Server Test URLs

Component	Managed Server Test URL
SOA	http://*vip*:*port*/soa-infra
OIM	http://*vip*:*port*/identity

14. Configure Server Migration for the new Managed Server.

> **Note:**
>
> Because this new node is using an existing shared storage installation, it is already using a Node Manager and environment configured for server migration that includes netmask, interface, wlsifconfig script superuser privileges. The floating IP for the new Managed Server is already in the new node.

To configure server migration:

a. Log into the Administration Console.

b. In the left pane, expand **Environment** and select **Servers**.

c. Select the server (represented as a hyperlink) for which you want to configure migration. The Settings page for that server appears.

d. Click the Migration tab.

e. In the Available field, in the Migration Configuration section, select machines to which to enable migration and click the right arrow.

> **Note:**
>
> Specify the least-loaded machine as the new server's migration target. Required capacity planning must be completed so that this node has the available resources to sustain an additional Managed Server.

f. Select the **Automatic Server Migration Enabled** option. This enables the Node Manager to start a failed server on the target node automatically.

g. Click **Save**.

h. Restart the Administration Server, Managed Servers, and Node Manager.

15. Test server migration for this new server from the node where you added it:

    a. Stop the Managed Server.

    Run `kill -9` *pid* on the PID of the Managed Server. Identify the PID of the node using, for example, `ps -ef | grep WLS_SOA`*n*.

    b. Watch the Node Manager Console for a message indicating that the floating IP has been disabled.

    c. Wait for the Node Manager to try a second restart of the new Managed Server. Node Manager waits for a fence period of 30 seconds before restarting.

    d. After Node Manager restarts the server, stop it again. Node Manager should log a message that the server will not restart again locally.

16. Edit the OHS configuration file to add the new managed server(s). See Configuring Oracle HTTP Server to Recognize New Managed Servers.

- Configuring Oracle HTTP Server to Recognize New Managed Servers

Configuring Oracle HTTP Server to Recognize New Managed Servers

To complete scale up/scale out, you must edit the `oim.conf` file to add the new Managed Servers, then restart the Oracle HTTP Servers.

1. Go to the directory *ORACLE_INSTANCE*`/config/OHS/component/moduleconf`
2. Edit `oim.conf` to add the new Managed Server to the WebLogicCluster directive. You must take this step for each URLs defined for OIM or SOA. Each product must have a separate `<Location>` section. Also, ports must refer to the Managed Servers. For example:

```
<Location /oim
    SetHandler weblogic-handler
    WebLogicCluster
 host1.example.com:14200,host2.example.com:14200
</Location>
```

3. Restart Oracle HTTP Server on WEBHOST1 and WEBHOST2:

```
WEBHOST1>opmnctl stopall
WEBHOST1>opmnctl startall

WEBHOST2>opmnctl stopall
WEBHOST2>opmnctl startall
```

> **Note:**
>
> If you are not using shared storage system (Oracle recommended), copy oim.conf to the other OHS servers.

> **Note:**
>
> See the General Parameters for WebLogic Server Plug-Ins in *Oracle Fusion Middleware Using Web Server 1.1 Plug-Ins with Oracle WebLogic Server* for additional parameters that can facilitate deployments.

11

Configuring High Availability for Oracle Access Manager Components

An introduction to Oracle Access Manager and description of how to design and deploy a high availability environment for Access Manager.

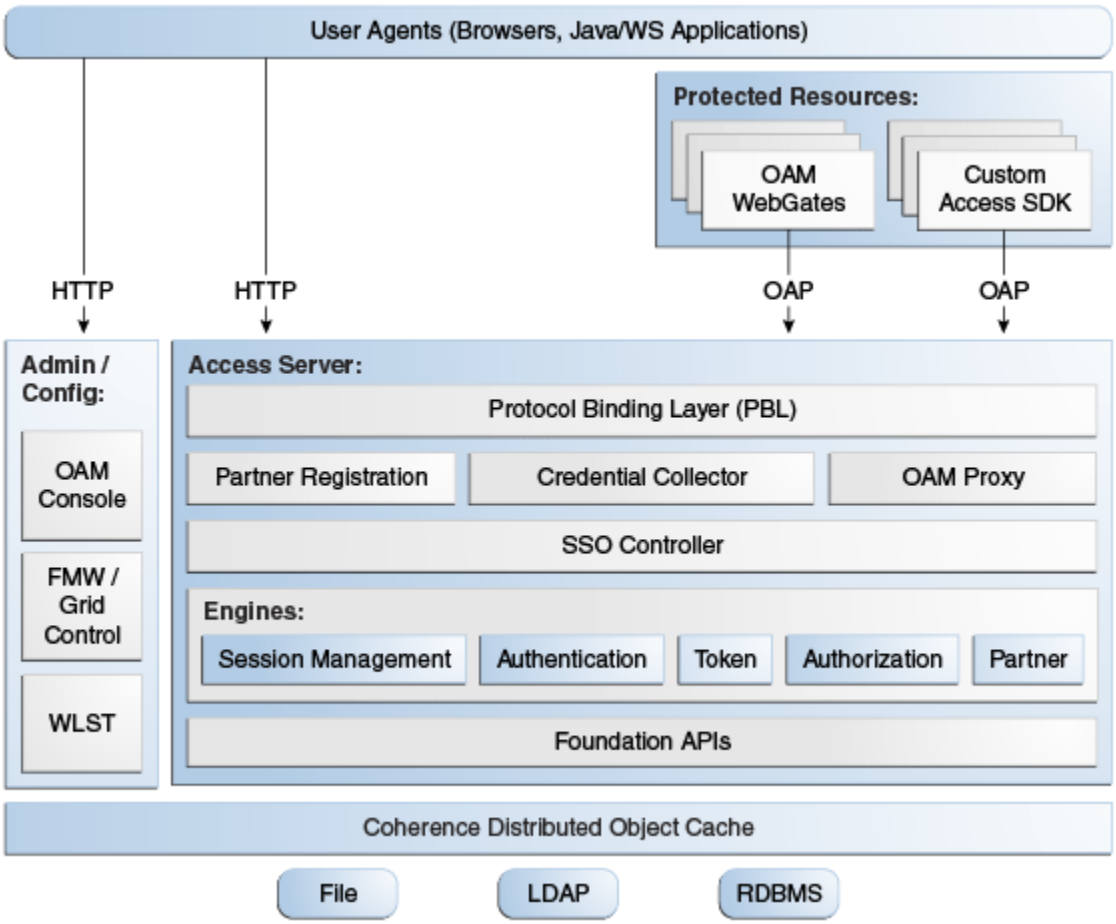
Access Manager provides a single authoritative source for all authentication and authorization services. See Introduction to Oracle Access Manager in *Oracle Fusion Middleware Administrator's Guide for Oracle Access Management*.

Access Manager Component Architecture

An introduction to primary Access Manager components and architecture.

Figure 11-1 shows the Access Manager component architecture.

Figure 11-1 Access Manager Single Instance Architecture



Following are the components discussed in the Access Manager Single Instance Architecture:

- **User agents**: Include web browsers, Java applications, and Web services applications. User agents access the Access Server and administration and configuration tools using HTTP.
- **Protected resources**: Application or web page to which access is restricted. WebGates or Custom Agents control access to protected resources.
- **Administration and configuration tools**: Administer and configure Access Manager with Oracle Access Management Console, Oracle Enterprise Manager Fusion Middleware Control and Oracle Enterprise Manager Grid Control, and WebLogic Scripting Tool (WLST).
- **Access Server**: Includes Credential Collector and OAM Proxy components.
- Access Manager Component Characteristics
  A typical Access Manager deployment consists of system entities such as user agents, protected resources, and access server.
- Access Manager Configuration Artifacts
- Access Manager External Dependencies

Access Manager Component Characteristics

A typical Access Manager deployment consists of system entities such as user agents, protected resources, and access server.

A list of system entities and the characteristics required for an Access Manager deployment:

- **Access Manager Agents** - Access Server extensions that ensure access is controlled according to policies that Access Server manages. Agents require the Access Server component to perform their functions. If Access Server is unavailable, access to protected servers is denied; users are locked out of the system.
- **Protected Resources** (partnered applications) - Applications that Access Manager protects. Access to these resources depends on access control policies in Access Manager and enforced by Access Manager agents deployed in the protected resource's access path.
- **Access Server** - Server side component. Provides core runtime access management services.
- **JMX Mbeans** - Runtime Mbeans are packaged as part of the Access Server package. Config Mbeans are packaged as standalone WAR files.
- **WebLogic 12*c* SSPI providers** consist of Java classes that implement the SSPI interface along with Access Java Access JDK. AccessGates are built using pure Java Access JDK.
- **Oracle Access Management Console** - Application that hosts Administration Console and provides services to manage Access Manager deployment.
- **WebLogic Scripting Tool** - Java classes included in Access Server package. Limited administration of Access Manager deployment is supported via the command line.
- **Fusion Middleware Control and Enterprise Manager Grid Control** - Access Manager integrates with Enterprise Manager Grid Control to show performance metrics and deployment topology.
- **Access Manager Proxy** - Custom version of Apache MINA server. Includes MessageDrivenBeans and ResourceAdapters in addition to Java Server classes.
- **Data Repositories** - Access Manager handles different types of information including Identity, Policy, Partner, Session and Transient data:
  - LDAP for Identity data
  - Files for Configuration and Partner data
  - Policy data will be stored in files or in an RDBMS
- Oracle **Access Manager WebGates** are C-based agents that are intended to be deployed in web servers.
- **Oracle Single Sign-On Apache** modules are C-based agents that are intended to be deployed in Oracle HTTP Server web servers.

Access Manager Configuration Artifacts

Access Manager configuration artifacts include:

Table 11-1 Access Manager Configuration Artifacts

Configuration Artifact	Description
DOMAIN_HOME/config/fmwconfig/oam-config.xml	Configuration file which contains instance specific information.
DOMAIN_HOME/config/fmwconfig/oam-policy.xml	Policy store information.
DOMAIN_HOME/config/fmwconfig/.oamkeystore	Stores symmetric and asymmetric keys.
DOMAIN_HOME/config/fmwconfig/component_events.xml	Used for audit definition.
DOMAIN_HOME/config/fmwconfig/jazn-data.xml	Administration Console permissions
DOMAIN_HOME/config/fmwconfig/servers/*instanceName*/logging.xml	Logging configuration. Do not edit this file manually.
DOMAIN_HOME/config/fmwconfig/servers/*instanceName*/dms_config.xml	Tracing logging. Do not edit this file manually.
DOMAIN_HOME/config/fmwconfig/cwallet.sso	Stores passwords that OAM uses to connect to identity stores, database, and other entities. This is not for end user passwords.
DOMAIN_HOME/output	Stores agent configuration files.

Access Manager External Dependencies

The following table describes Access Manager external runtime dependencies.

Table 11-2 Access Manager External Dependencies

Dependency	Description
LDAP based Identity Store	• User Identity Repository • LDAP access abstracted by User/Role API. Access Manager always connects to one Identity store: a physical server or a load balancer IP. If the primary down, Access Manager reconnects and expects the load balancer to connect it to the secondary.
OCSP Responder Service	Real-time X.509 Certification Validation
RDBMS Policy Store	• Policy (Authentication and Authorization) Repository • RDBMS access abstracted by the OAM policy engine
Oracle Identity Manager Policy Store (when Oracle Identity Manager-based password management is enabled)	LDAP Repository containing Oblix Schema elements that are used to store Configuration, Metadata, and so on
Identity Federation	Dependency when Identity Federation Authentication Scheme is selected
OCSP Responder Service	Real-time X.509 Certification Validation

• Access Manager Log File Location

Access Manager Log File Location

You deploy Access Manager on WebLogic Server. Log messages go to the server log file of the WebLogic Server that you deploy it on. The default server log location is:

```
Domain_HOME/servers/serverName/logs/ serverName-diagnostic.log
```

Access Manager High Availability Concepts

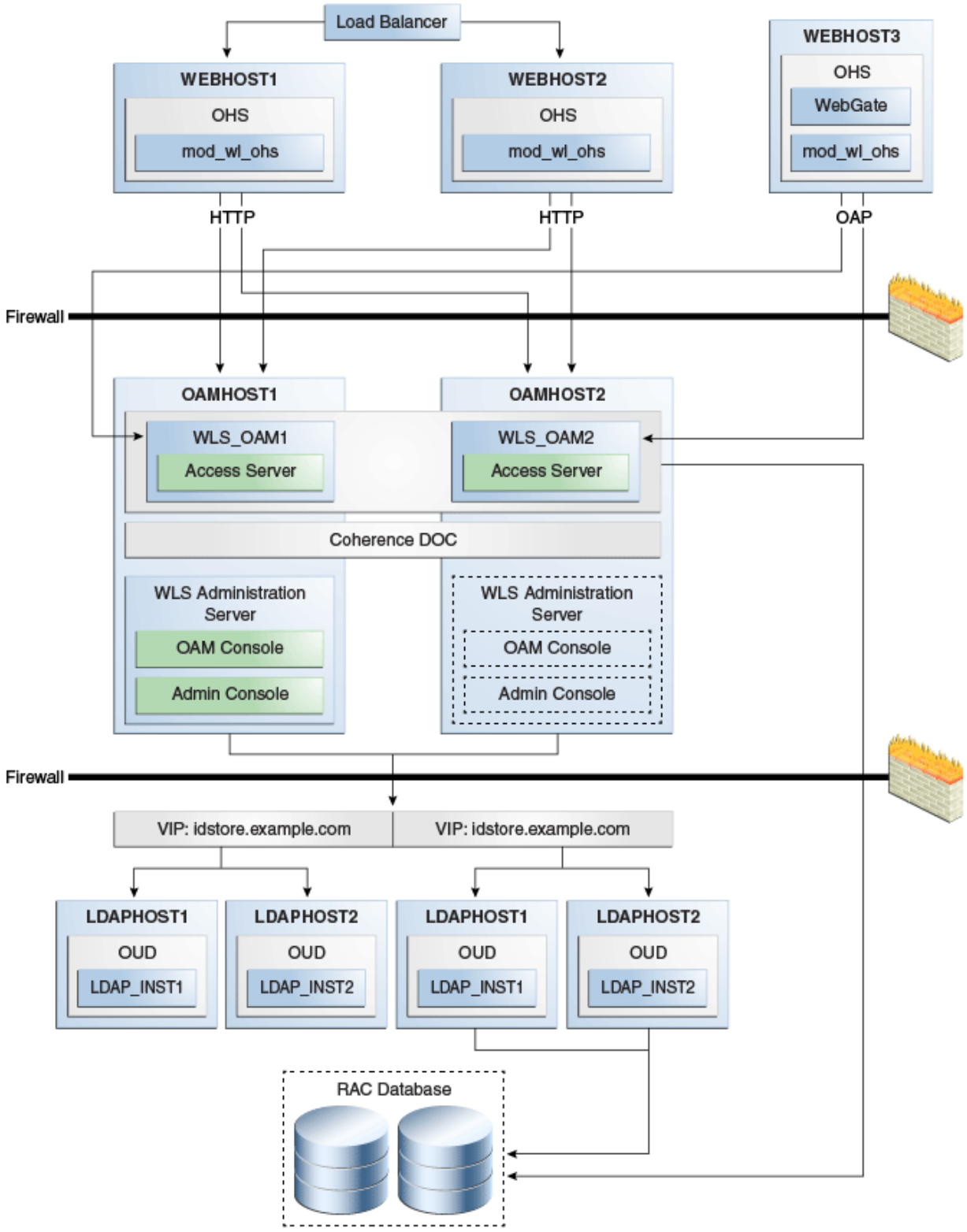
This following sections provide conceptual information about using Access Manager in a high availability two-node cluster.

- Access Manager High Availability Architecture
- Protection from Failures and Expected Behaviors

Access Manager High Availability Architecture

Figure 11-2 shows an Access Manager high availability architecture:

Figure 11-2 Access Manager High Availability Architecture



In Figure 11-2, the hardware load balancer receives incoming authentication requests and routes them to WEBHOST1 or WEBHOST2 in the web tier. These hosts have

Oracle HTTP Server installed. Oracle HTTP Server then forwards requests on to the WebLogic managed servers using the WebLogic plugin `mod_wl_ohs.conf`. See Oracle HTTP Server Configuration.

The load balancing router should use session stickiness for HTTP traffic only. OAP traffic does not use a load balancing router, so session stickiness is not required for OAP traffic.

Applications that other Oracle HTTP Servers access, that in turn have resources with restricted access, must have a WebGate and a custom agent configured. The WebGate on WEBHOST3 communicates with the Access Servers on OAMHOST1 and OAMHOST2 in the application tier using OAP. WEBHOST3 is an application web server, and for authentication, HTTP redirect routes requests to the load balancer and WEBHOST1 and WEBHOST2. For a high availability deployment, you can configure another host (for example, WEBHOST4) with the same components as WEBHOST3.

OAMHOST1 and OAMHOST2 deploy managed servers which host the Oracle Access Server application. These managed servers are configured in a cluster which enables the Access Servers to work in an active-active manner.

The Administration Server runs on OAMHOST1 and deploys the WebLogic Administration Console, Oracle Enterprise Manager Fusion Middleware Control, and the Oracle Access Management Console.

In the directory tier, the virtual IP `idstore.example.com` routes the IDstore requests to LDAPHOST1 and LDAPHOST2, which comprise an active-active IDStore cluster. For example the virtual IP `oud.example.com` is set up to route Oracle Unified Directory requests to OUDHOST1 and OUDHOST2, which comprise an active-active Oracle Unified Directory cluster.

An Oracle RAC database provides high availability in the data tier. The Oracle RAC database is configured in a JDBC multi data source or GridLink data source to protect the instance from Oracle RAC node failure.

In Access Manager 12*c*, only one Access Manager cluster is supported per WebLogic Server domain. Access Manager clusters cannot span WebLogic Server domains.

A single instance Access Manager deployment satisfies the following high availability requirements:

- Load handling
- External connection management and monitoring
- Recovery
- Fault containment
- Fault diagnostics
- Administration Server offline

A multiple instance Access Manager deployment satisfies the following additional high availability requirements:

- Redundancy
- Client connection failover/continuity
- Client load balancing
- State management

Oracle recommends using an external load balancing router for inbound HTTP connections. Outbound external connections to LDAP Servers (or OAM policy engine PDP/PIP) are load balanced with support for connection failover. Therefore, a load balancer is not required. Access Manager agents, typically WebGates, can load balance connections across multiple Access Servers.

Access Manager agents open persistent TCP connections to the Access Servers. This requires firewall connection timeouts to be sufficiently large to avoid premature termination of TCP connections.

The Access Server and Access Manager Administration Console interface with the OAM policy engine for policy evaluation and management. The OAM policy engine internally depends on a database as the policy repository. The database interactions are encapsulated within the OAM policy engine, with only the connectivity configuration information managed by Access Manager. The high availability characteristics of the interaction between Access Manager and the OAM policy engine are:

- The database connection information is configured in the Access Manager configuration file synchronized among the Access Manager instances.
- Database communication is managed within the OAM policy engine, and generally decoupled from Access Manager and OAM policy engine interactions. The very first startup of an OAM server instance will fail, however, if the database is unreachable. An OAM policy engine bootstrap failure is treated as fatal by Access Manager, and the startup operation is aborted.
- Access Manager policy management interfaces (in the Oracle Access Management Console and the CLI tool) fail if the database is unreachable, as seen by the OAM policy engine management service interfaces. The operation may be retried at a later point in time, but no automated retry is provided for management operations.
- Following a successful policy modification in the database repository, the OAM policy engine layer in the OAM server runtimes retrieves and activates the changes within a configurable OAM policy engine database poll interval (configured through Access Manager configuration). A positive acknowledgement of a policy change must be received from each OAM server runtime, otherwise the policy change cannot be considered successfully activated. The administrator can use the Oracle Access Management Console to remove any Access Manager instance with a policy activation failure from service.

Protection from Failures and Expected Behaviors

The WebLogic Server infrastructure protects the Identity Management Service Infrastructure system from all process failures. These features protect an Access Manager high availability configuration from failure

- Back channel OAP bindings use a primary/secondary model for failover. Front Channel HTTP bindings use a load balancing router for failover.
- If an Access Server fails, a WebGate with a persistent connection to that server waits for the connection to timeout, then it switches over to the secondary (backup) Access Server. Outstanding requests fail over to the secondary server.
- Access Manager Access Servers support a heartbeat check. Also, the WebLogic Node Manager on the Managed Server can monitor the application and restart it.

- If a WebLogic Server node fails, external connection failover is based on the configuration, the retry timeout, and the number of retries. Access Manager Agent-Access Server failover is based on a timeout.
- If the load balancing router or proxy server detects a WebLogic Server node failure, subsequent client connections route to the active instance, which picks up the session state and carries on with processing.
- When the lifetime of a connection expires, pending requests complete before the connection terminates. The connection object returns to the pool.
- When it receives an exception from another service, Access Manager retries external connection requests. You can configure the number of retries.
- WebLogic Server Crash
- Node Failure
- Database Failure

WebLogic Server Crash

If a Managed Server fails, Node Manager attempts to restart it locally

Ongoing requests from Oracle HTTP Server timeout and new requests are directed to the other Managed Server. After the server's restart completes on the failed node, Oracle HTTP Server resumes routing any incoming requests to the server.

> **Note:**
>
> Access Manager servers support a heartbeat check to determine if the access server can service its requests. It checks:
>
> - Whether the LDAP store can be accessed
> - Whether the policy store can be accessed
>
> If the heartbeat succeeds, the Access Server can service requests and requests are sent to it. If the heartbeat fails, requests do not route to the Access Server.

Node Failure

Node failures are treated in the same way as WebLogic Server fails.

Database Failure

Multi data sources protect Access Manager service Infrastructure against failures. When an Oracle RAC database instance fails, connections are reestablished with available database instances. The multi data source enables you to configure connections to multiple instances in an Oracle RAC database.

For more on multi data source configuration, see Section 4.1.3, "Using Multi Data Sources with Oracle RAC".

High Availability Directory Structure Prerequisites

A high availability deployment requires product installations and files to reside in specific directories. A standard directory structure facilitates configuration across nodes and product integration.

The following table describes high availability directory structure prerequisites.

Table 11-3 Directory Structure Prerequisites

Directory	Requirements
ORACLE_HOME	**Each product must have its own ORACLE_HOME**. For example, OAM and OIM must go in separate *ORACLE_HOME* locations. **ORACLE_HOME contents must be identical across all nodes.** Across all nodes, *ORACLE_HOME* must: • Reside in the file system at the same path • Contain identical products • Contain identical versions of those products • Have identical *ORACLE_HOME* names • Have identical patches installed
DOMAIN_HOME and *APPLICATION_DIRECTORY*	These directories must have the same path on all nodes. Put these directories in a separate file system location from *ORACLE_HOME*; do not put these directories in the *ORACLE_HOME*/`user_projects` directory
`wlsserver_10.`*n*	Each OAM and OIM installation requires its own, separate WebLogic Server installation.

You have three options to set up the high availability directory structure:

- Use shared storage to store *ORACLE_HOME* directories. Oracle recommends this option. Use a NFS exported by a NAS, or a cluster file system pointing to a SAN/NAS.
- Use local storage and run all installation, upgrade, and patching procedures on one node, then replicate to other nodes (using rsync, for example.)
- Use local storage and repeat all installation and patch procedures on each node.

Access Manager High Availability Configuration Steps

This section provides high-level instructions to set up a high availability deployment for Access Manager. This deployment includes two Oracle HTTP Servers, which distribute requests to two OAM servers. These OAM servers interact with an Oracle Real Application Clusters (Oracle RAC) database and, optionally, an external LDAP store. If any single component fails, the remaining components continue to function.

See Using Dynamic Clusters.

- Access Manager Configuration Prerequisites
- Running the Repository Creation Utility to Create the Database Schemas
- Installing Oracle WebLogic Server
- Installing and Configuring the Access Manager Application Tier

Access Manager Configuration Prerequisites

Before you configure Access Manager for high availability, you must:

- Install Oracle WebLogic Server on OAMHOST1 and OAMHOST2. See Installing Oracle WebLogic Server.
- Install the Oracle Identity Management executables on OAMHOST1 and OAMHOST2. See the Installing and Configuring the Access Manager Application Tier.
- Run the Repository Creation Utility to create the Access Manager schemas in a database. See Running the Repository Creation Utility to Create the Database Schemas.
- Ensure that a highly available LDAP implementation is available.

For example,

- Install the Infrastructure jar, `jdk8/bin/java -jar fmw_12.2.1.3.0_infrastructure_generic.jar` and change the default installation directory path manually from `/tmp/Middleware/ORACLE_HOME` to `/tmp/Middleware/`
- Install IDM jar, `jdk8/bin/java -jar fmw_12.2.1.3.0_idm_generic.jar` and choose `/tmp/Middleware/` as the installation directory.
- Run RCU located at `/tmp/Middleware/oracle_common/bin/rcu`

Running the Repository Creation Utility to Create the Database Schemas

The schemas you create depend on the products you want to install and configure. See *Planning an Installation of Oracle Fusion Middleware* and *Creating Schemas with the Repository Creation Utility* to run RCU.

Installing Oracle WebLogic Server

To install Oracle WebLogic Server, see *Installing and Configuring Oracle WebLogic Server and Coherence*.

> **Note:**
>
> On 64-bit platforms, when you install Oracle WebLogic Server using the generic jar file, JDK is not installed with Oracle WebLogic Server. You must install JDK separately, before installing Oracle WebLogic Server.

Installing and Configuring the Access Manager Application Tier

See Installing and Configuring Identity and Access Management in *Installing and Configuring Oracle Identity and Access Management*.

Creating boot.properties for the Administration Server on OAMHOST1

The `boot.properties` file enables the Administration Server to start without prompting for the administrator's username and password.

To create the `boot.properties` file:

1. On OAMHOST1, go to:

   ORACLE_HOME`/user_projects/domains/`*domainName*`/servers/AdminServer/security`

   For example:

   ```
   cd /u01/app/oracle/product/fmw/user_projects/domains/IDMDomain/servers/
   AdminServer/security
   ```

2. Use a text editor to create a file called `boot.properties` under the `security` directory. Enter the following lines in the file:

   ```
   username=adminUser
   password=adminUserPassword
   ```

   > **Note:**
   >
   > When you start Administration Server, username and password entries in the file get encrypted. For security reasons, minimize the time the entries in the file are left unencrypted. After you edit the file, start the server as soon as possible to encrypt the entries.

3. Stop the Administration Server if it is running.

   See Starting and Stopping Oracle Fusion Middleware in *Administering Oracle Fusion Middleware* to start and stop WebLogic Servers.

4. Start Node Manager by using the following commands:

   ```
   cd WL_HOME/server/bin ./startNodeManager.sh
   ```

5. Start the Administration Server on OAMHOST1 with the `startWebLogic.sh` script in the *ORACLE_HOME*`/user_projects/domains/`*domainName*`/bin` directory.

6. Validate that changes are successful. Open a browser and log into these consoles using the `weblogic` user credentials:

- WebLogic Server Administration Console at:

  `http://oamhost1.example.com:7001/console`

- Oracle Enterprise Manager Fusion Middleware Control at:

  `http://oamhost1.example.com:7001/em`

Starting OAMHOST1

The following sections describe the steps for starting OAMHOST1.

- Start Node Manager
- Start Access Manager on OAMHOST1

Start Node Manager

- Start Node Manager by issuing the following command:

  ```
  OAMHOST1>ORACLE_HOME/user_projects/domains/domainName/bin/startNodeManager.sh
  ```

Start Access Manager on OAMHOST1

To start Access Manager on OAMHOST1, follow these steps:

1. Log into the WebLogic Administration Console using this URL using WebLogic administrator credentials:

   `http://oamhost1.example.com:7001/console`

2. Start the WLS_OAM1 Managed Server using the WebLogic Server Administration Console, as follows:
   a. Expand the **Environment** node in the **Domain Structure** tree on the left.
   b. Click **Servers**.
   c. On the Summary of Servers page, open the **Control** tab.
   d. Select **WLS_OAM1**, and then click **Start**.
   e. Click **YES** to confirm that you want to start the server.
   f. Then select **OAM_POLICY_MGR1**, and then click **Start**.
   g. Click **YES** to confirm that you want to start the server.

Validating OAMHOST1

Validate the implementation by connecting to the OAM server:

```
http://OAMHOST1.example.com:14150/access
http://OAMHOST1.example.com:14100/oam/server/logout
```

The implementation is valid if an OAM logout successful page opens.

Configuring OAM on OAMHOST2

After configuration succeeds on OAMHOST1, propagate it to OAMHOST2. Pack the domain using the `pack` script on OAMHOST1 and unpack it with the `unpack` script on OAMHOST2.

Both scripts reside in the *ORACLE_HOME*/oracle_common/common/bin directory.

On OAMHOST1, enter:

```
pack.sh -domain=$ORACLE_HOME/user_projects/domains/IDM_Domain \
    -template=/tmp/idm_domain.jar -template_name=OAM Domain -managed=true
```

This creates a file called idm_domain.jar in the /tmp directory. Copy this file to OAMHOST2.

On OAMHOST2, enter:

```
unpack.sh -domain=$ORACLE_HOME/user_projects/domains/IDM_Domain \
    -template=/tmp/idm_domain.jar
```

Starting OAMHOST2

This following sections describe the steps for starting OAMHOST2. Steps include the following:

- Create the Node Manager Properties File on OAMHOST2
- Start Node Manager
- Start Access Manager on OAMHOST2

Create the Node Manager Properties File on OAMHOST2

Before you can start managed servers from the console, you must create a Node Manager property file. Run the script setNMProps.sh, which is located in the *ORACLE_HOME*/oracle_common/common/bin directory. For example:

```
OAMHOST1> $ORACLE_HOME/oracle_common/common/bin/setNMProps.sh
```

Start Node Manager

Start Node Manager by issuing the following command:

```
OAMHOST2>ORACLE_HOME/user_projects/domains/domainName/bin/startNodeManager.sh
```

Start Access Manager on OAMHOST2

To start Access Manager on OAMHOST2:

1. Log into the WebLogic Administration Console using this URL:

   ```
   http://OAMHOST1.example.com:7001/console
   ```

2. Supply the WebLogic administrator username and password.
3. Select **Environment - Servers** from the **Domain Structure** menu.
4. Click the Control tab.
5. Click the server **WLS_OAM2**.
6. Click **Start**.
7. Click **OK** to confirm that you want to start the server.

Validating OAMHOST2

Validate the implementation by connecting to the OAM server:

```
http://OAMHOST2.example.com:14150/access
http://OAMHOST2.example.com:14100/oam/server/logout
```

The implementation is valid if an OAM logout successful page opens.

Configuring Access Manager to Work with Oracle HTTP Server

Complete the subsequent procedures to configure Access Manager to work with Oracle HTTP Server.

- Update Oracle HTTP Server Configuration
- Restart Oracle HTTP Server
- Make OAM Server Aware of the Load Balancer

Update Oracle HTTP Server Configuration

On WEBHOST1 and WEBHOST2, create a file named `oam.conf` in this directory:

```
OHSDomain/config/fmwconfig/components/OHS/<instancename>/moduleconf/
```

Create the file and add the following lines:

```
NameVirtualHost *:7777
<VirtualHost *:7777>

    ServerName login.example.com:7777
    ServerAdmin you@your.address
    RewriteEngine On
    RewriteOptions inherit

    <Location /oam>
        SetHandler weblogic-handler
        Debug ON
        WLLogFile /tmp/weblogic.log
        WLProxySSL ON
        WLProxySSLPassThrough ON
        WLCookieName OAMSESSIONID
        WebLogicCluster OAMHOST1.example.com:14100,OAMHOST2.example.com:14100
    </Location>

    <Location /oamfed>
        SetHandler weblogic-handler
        Debug ON
        WLLogFile /tmp/weblogic.log
        WLProxySSL ON
        WLProxySSLPassThrough ON
        WLCookieName OAMSESSIONID
        WebLogicCluster OAMHOST1.example.com:14100,OAMHOST2.example.com:14100

    </Location>

    <Location /sts>
        SetHandler weblogic-handler
```

```
        WLLogFile /tmp/weblogic.log
        WLProxySSL ON
        WLProxySSLPassThrough ON
        WLCookieName OAM_JSESSIONID
        WebLogicCluster OAMHOST1.example.com:14100,OAMHOST2.example.com:14100

    </Location>

    <Location /access>
        SetHandler weblogic-handler
        Debug ON
        WLLogFile /tmp/weblogic.log
        WLProxySSL ON
        WLProxySSLPassThrough ON
        WLCookieName OAMSESSIONID
        WebLogicCluster amahost1.example.com:14150,amahost2.example.com:14150


    <Location /oamsso>
        SetHandler weblogic-handler
        Debug ON
        WLLogFile /tmp/weblogic.log
        WLProxySSL ON
        WLProxySSLPassThrough ON
        WLCookieName OAMSESSIONID
        WebLogicCluster oam_policy_mgr1.example.com:14100,oam_policy_
         mgr2.example.com:14100

    </Location>

</VirtualHost>
```

Restart Oracle HTTP Server

Restart the Oracle HTTP Server on WEBHOST1:

```
OHSDomain/bin/stopComponent.sh ohs1
OHSDomain/bin/startComponent.sh ohs1
```

Restart the Oracle HTTP Server on WEBHOST2:

```
OHSDomain/bin/stopComponent.sh ohs2
OHSDomain/bin/startComponent.sh ohs2
```

Make OAM Server Aware of the Load Balancer

By default, Access Manager sends requests to the login page on the local server. In a high availability deployment, you must change this setup so that login page requests go to the load balancer.

To make Access Manager aware of the load balancer:

1. Log into the Oracle Access Management Console at this URL as the `weblogic` user:

   ```
   http://OAMHOST1.example.com:7001/oamconsole
   ```

2. Click on the **Configuration** tab.
3. Click the **Access Manager Settings** link.

4. Enter the following information:
   - OAM Server Host: login.example.com
   - OAM Server Port: 7777
   - OAM Server Protocol: https
5. Click **Apply**.
6. Restart managed servers WLS_OAM1 and WLS_OAM2.

Configuring Access Manager to use an External LDAP Store

By default, Access Manager uses its own in built-in LDAP server. In a highly available environment, Oracle recommends an external LDAP directory as the directory store.

> **Note:**
>
> Oracle recommends that you back up the environment and LDAP store before following this procedure.

- Extending Directory Schema for Access Manager
- Creating Users and Groups in LDAP
- Creating a User Identity Store
- Setting LDAP to System and Default Store
- Setting Authentication to Use External LDAP
- Adding LDAP Groups to WebLogic Administrators
  Access Manager requires access to MBeans stored within the administration server. In order for LDAP users to be able to log in to the WebLogic console and Fusion Middleware control, they must be assigned the WebLogic Administration rights. In order for Access Manager to invoke these Mbeans, users in the IAMAdministrators group must have WebLogic Administration rights.

Extending Directory Schema for Access Manager

Pre-configuring the Identity Store extends the schema in the backend directory regardless of directory type.

To extend the directory schema for Access Manager, perform these steps on OAMHOST1:

1. Set the Environment Variables: `JAVA_HOME`, `IDM_HOME` and `ORACLE_HOME`.

   Set `IDM_HOME` to `IDM_ORACLE_HOME`

   Set `ORACLE_HOME` to `IAM_ORACLE_HOME`
2. Create a properties file `extend.props` that contains the following:

```
IDSTORE_HOST : idstore.example.com

IDSTORE_PORT : 389

IDSTORE_BINDDN : cn=orcladmin
```

```
IDSTORE_USERNAMEATTRIBUTE: cn

IDSTORE_LOGINATTRIBUTE: uid

IDSTORE_USERSEARCHBASE:cn=Users,dc=example,dc=com

IDSTORE_GROUPSEARCHBASE: cn=Groups,dc=us,dc=oracle,dc=com

IDSTORE_SEARCHBASE: dc=example,dc=com

IDSTORE_SYSTEMIDBASE: cn=systemids,dc=example,dc=com
```

Where:

- `IDSTORE_HOST` and `IDSTORE_PORT` are, respectively, the host and port of your Identity Store directory. Specify the back-end directory here rather than OUD.)
- `IDSTORE_BINDDN` Administrative user in the Identity Store Directory
- `IDSTORE_USERSEARCHBASE` Location in your Identity Store where users are placed.
- `IDSTORE_GROUPSEARCHBASE` Location in your Identity Store where groups are placed.
- `IDSTORE_SEARCHBASE` Location in the directory where Users and Groups are stored.
- `IDSTORE_SYSTEMIDBASE` Location in your directory where the Oracle Identity Manager reconciliation users are placed.
- `IDSTORE_SYSTEMIDBASE` Location of a container in the directory where you can place users when you do not want them in the main user container. This happens rarely. For example, if Oracle Identity Manager reconciliation user which is also used for the bind DN user in Oracle Virtual Directory adapters.

3. Configure the Identity Store using the command `idmConfigTool`, located at `IAM_ORACLE_HOME/idmtools/bin`.

   The command syntax is:

   ```
   idmConfigTool.sh -preConfigIDStore input_file=configfile
   ```

   For example:

   ```
   idmConfigTool.sh -preConfigIDStore input_file=extend.props
   ```

   The system prompts you for the account password with which you are connecting to the Identity Store.

   Sample command output:

   ```
   Enter ID Store Bind DN password :
   Apr 5, 2011 3:39:25 AM oracle.ldap.util.LDIFLoader loadOneLdifFile
   INFO: -> LOADING:


   /u01/app/oracle/product/fmw/IAM/idmtools/templates/oid/
   idm_idstore_groups_template.ldif
   Apr 5, 2011 3:39:25 AM oracle.ldap.util.LDIFLoader loadOneLdifFile
   INFO: -> LOADING:
   ```

```
/u01/app/oracle/product/fmw/IAM/idmtools/templates/oid/
idm_idstore_groups_acl_template.ldif
Apr 5, 2011 3:39:25 AM oracle.ldap.util.LDIFLoader loadOneLdifFile
INFO: -> LOADING:


/u01/app/oracle/product/fmw/IAM/idmtools/templates/oid/systemid_pwdpolicy.ldif
Apr 5, 2011 3:39:25 AM oracle.ldap.util.LDIFLoader loadOneLdifFileINFO: -> LOADING:


/u01/app/oracle/product/fmw/IAM/idmtools/templates/oid/idstore_tuning.ldifApr 5,
2011 3:39:25 AM oracle.ldap.util.LDIFLoader loadOneLdifFileINFO: -> LOADING:


/u01/app/oracle/product/fmw/IAM/idmtools/templates/oid/oid_schema_extn.ldif
The tool has completed its operation. Details have been logged to automation.log
```

4. Check the log file for errors and warnings and correct them.

Creating Users and Groups in LDAP

To add users that Access Manager requires to the Identity Store, follow these steps:

1. Set the Environment Variables `JAVA_HOME`, `IDM_HOME`, and `ORACLE_HOME`.
   - Set `IDM_HOME` to `IDM_ORACLE_HOME`.
   - Set `ORACLE_HOME` to `IAM_ORACLE_HOME`.
2. Create a properties file `oam.props` that contains the following parameters shown in the following example:

```
IDSTORE_HOST: host.example.com
IDSTORE_PORT: 9389
IDSTORE_BINDDN: cn=directory manager
IDSTORE_PASSWD: secret12
IDSTORE_USERNAMEATTRIBUTE: cn
IDSTORE_LOGINATTRIBUTE: uid
IDSTORE_USERSEARCHBASE: ou=people,dc=example,dc=com
IDSTORE_GROUPSEARCHBASE: ou=groups,dc=example,dc=com
IDSTORE_SEARCHBASE: dc=example,dc=com
IDSTORE_SYSTEMIDBASE: cn=systemids,dc=example,dc=com
IDSTORE_OAMSOFTWAREUSER: oamSoftwareUser
IDSTORE_PWD_OAMSOFTWAREUSER: example_pwd
IDSTORE_OAMADMINUSER: oamAdminUser
IDSTORE_PWD_OAMADMINUSER: example_pwd
IDSTORE_PWD_OBLIXANONYMOUSUSER: example_pwd
IDSTORE_PWD_ANONYMOUSUSER: example_pwd
OAM11G_IDSTORE_ROLE_SECURITY_ADMIN: OAMAdministrators
POLICYSTORE_SHARES_IDSTORE: true
```

3. Configure the Identity Store using the command `idmConfigTool` which is located at `IAM_ORACLE_HOME/idmtools/bin`.

The command syntax is as shown in the following example:

```
$ORACLE_HOME/idmtools/bin/idmConfigTool.sh -prepareIDStore mode=OAM
input_file=prepareIDStore.properties log_level=ALL
log_file=log_idstore1.out dump_params=true
```

After the command runs, the system prompts you to enter the password for the account with which you are connecting to the ID Store.

4. Check the log file for any errors or warnings and correct them.

Creating a User Identity Store

To create a user identity store:

1. Go to the Oracle Access Management Console at the URL:

   ```
   http://adminvhn.example.com:7001/oamconsole
   ```

2. Log in using the WebLogic administration user.
3. Select **Configuration** tab and click **User Identity Stores**.
4. Under OAM ID Stores, click **Create**. Enter the following information:
   - **Store Name**: LDAP_DIR
   - **Store Type**: OUD
   - **Description**: Enter a description of the Directory Store
   - **Enable SSL**: Select this if you communicate with your directory over SSL
   - **Location**: Enter the location, for example oud.example.com:389
   - **Bind DN**: Enter the user permitted to search the LDAP store. For example, cn=orcladmin
   - **Password**: Enter the oracleadmin password
   - **User Name Attribute**: For example: uid
   - **User Search Base**: Enter the location of users in the LDAP store. For example, cn=Users,dc=example,dc=com
   - **Group Name Attribute**: For example: orclguid
   - **Group Search Base**: Enter the location of groups in the LDAP store. For example, cn=Groups,dc=example,dc=com
   - **OAM Administrator Role**: OAMAdministrators
5. Click **Apply**.
6. Click **Test Connection** to validate the connection to the LDAP server.

Setting LDAP to System and Default Store

After you define the LDAP identity store, you must set it as the primary authentication store. Follow these steps in the Oracle Access Management Console:

1. From the **Configuration** tab, click **User Identity Stores**.
2. Select **LDAP_DIR** as Default Store.

3. Select **LDAP_DIR** as System Store.
4. Click the Add **[+]** icon in **Access System Administrators**.
5. Enter **OAM*** in the search name field and click **Search**.
6. Select **OAMAdministrators** from the search results and click **Add Selected**.
7. Click **Apply**.
8. In the Validate System Administrator window, enter the username and password of the OAM administrator, for example, oamadmin.
9. Click **Validate**.
10. Test the connection by clicking **Test Connection**.

Setting Authentication to Use External LDAP

By default, Access Manager uses the integrated LDAP store for user validation. You must update the LDAP authentication module so that it can validate users against the new external LDAP store.

To update the LDAP authentication module to use external LDAP:

1. Under **Application Security** tab, select **Authentication Modules** and click **Search**.
2. Click **LDAP**.
3. Select **Open** from the **Actions** menu.
4. Set **User Identity Store** to `LDAP_DIR`.
5. Click **Apply**.
6. Restart the Managed Servers Admin Server, WLS_OAM1 and WLS_OAM2.

Adding LDAP Groups to WebLogic Administrators

Access Manager requires access to MBeans stored within the administration server. In order for LDAP users to be able to log in to the WebLogic console and Fusion Middleware control, they must be assigned the WebLogic Administration rights. In order for Access Manager to invoke these Mbeans, users in the IAMAdministrators group must have WebLogic Administration rights.

When Single Sign-on is implemented, provide the LDAP group IDM Administrators with WebLogic administration rights, so that you can log in using one of these accounts and perform WebLogic administrative actions.

To add the LDAP Groups `IAMAdministrators` and `WLSAdmins` to the WebLogic Administrators:

1. Log in to the WebLogic Administration Server Console.
2. In the left pane of the console, click **Security Realms**.
3. On the Summary of Security Realms page, click **myrealm** under the Realms table.
4. On the Settings page for **myrealm**, click the **Roles & Policies** tab.
5. On the Realm Roles page, expand the **Global Roles** entry under the Roles table.
6. Click the **Roles** link to go to the Global Roles page.
7. On the Global Roles page, click the **Admin** role to go to the Edit Global Roles page.

8. On the Edit Global Roles page, under the **Role Conditions** table, click the **Add Conditions** button.
9. On the Choose a Predicate page, select **Group** from the drop down list for predicates and click **Next**.
10. On the Edit Arguments Page, Specify **IAMAdministrators** in the **Group Argument** field and click **Add**.
11. Repeat for the Group **WLSAdmins**.
12. Click **Finish** to return to the Edit Global Roles page.
13. The **Role Conditions** table now shows the groups **IAMAdministrators and WLSAdmins** as role conditions.
14. Click **Save** to finish adding the Admin role to the OAMAdministrators and IDM Administrators Groups.

Validating the Access Manager Configuration

Validate the configuration by logging into the Oracle Access Management Console at `http://OAMHOST1.example.com:7001/oamconsole` as `oamadmin`.

See Adding LDAP Groups to WebLogic Administrators

Scaling Up Access Manager Topology

You *scale up* to add a new Access Manager managed server to a node already running one or more server instances.

Scaling Up Access Manager

To scale up OAM:

1. Log in to the Administration Console at `http://hostname.example.com:7001/console`. From the Domain Structure window, expand the **Environment** node and then **Servers**.
2. In the Change Center, click **Lock & Edit**.
3. Select a server on the host you want to extend, for example: `WLS_OAM1`.
4. Click **Clone**.
5. Enter the following information:
   - **Server Name**: A new name for the server, for example: `WLS_OAM3`.
   - **Server Listen Address**: The name of the host on which the managed server will run.
   - **Server Listen Port**: The port the new managed server will use, this port must be unique within the host.
6. Click **OK**.

7. Click on the newly created server **WLS_OAM3**
8. Set the SSL listen port. This should be unique on the host that the managed server will run on.

> **Note:**
>
> Enable the SSL listen port 14101.

9. Click **Save**.
10. Disable hostname verification for the new managed server. You must do this before you start and verify the `WLS_OAM3` Managed Server. You can re-enable it after you configure server certificates for the communication between the Administration Server and Node Manager in `OAMHOSTn`.

    If the source server from which the new one was cloned had already disabled hostname verification, you do not need to take this step because the hostname verification settings propagated to the cloned server.

    To disable hostname verification, set **Hostname Verification** to `None` then click **Save**.
11. Click **Activate configuration** from the Change Center menu.

Registering the New Managed Server

To configure the new managed server as an OAM server, use the Oracle Access Management Console:

1. Log in to the Oracle Access Management Console as the `oamadmin` user at `http://oamhost1.example.com:7001/oamconsole`
2. Click the **Configuration** tab. Click **Server Instances**.
3. Select **Create** from the Actions menu.
4. Enter the following information:
   - **Server Name**: `WLS_OAM3`
   - **Host**: Host that the server will run on
   - **Port**: Listen port that was assigned when the managed server was created, for example, 14100
   - **Proxy Server ID**: `AccessServerConfigProxy`
   - **Port**: Port you want the OAM proxy to run on. This is unique for the host, for example, 5575
   - **Mode**: `Open`
5. Click **Apply** when a prompt requests that you confirm the edit.
6. Edit `oam-config.xml` available at `<DOMAIN_HOME>/config/fmwconfig` to set the IP range of nodes that get added dynamically.

The Settings Map for the following example is `SetWellKnownAddress>AuthorizedSubnets >Range1 >From [value of start ip range]]>To [value of end ip range].`

```
<Setting Name="AuthorizedSubnets" Type="htf:map">
<Setting Name="Range1" Type="htf:map">
<Setting Name="From" Type="htf:map">
<Setting Name="Key"
Type="xsd:string">oam.coherence.auth.range.from.1</Setting>
<Setting Name="Value"
Type="xsd:string">10.229.139.20</Setting>
</Setting>
<Setting Name="To" Type="htf:map">
<Setting Name="Key"
Type="xsd:string">oam.coherence.auth.range.to.1</Setting>
<Setting Name="Value"
Type="xsd:string">10.229.139.40</Setting>
</Setting>
</Setting>
</Setting>
```

7. Ensure that the `OAM_ORACLE_HOME` property `<ORACLE_HOME>/oam` is set while starting server nodes (OAM1, OAM2 etc). In this environment, `startWeblogic` script is edited to pass `-DOAM_ORACLE_HOME=<ORACLE_HOME>/oam` while starting the Java process.

Configuring WebGate with the New OAM Managed Server

To configure the WebGate with the new OAM Managed Server, take these steps:

1. Verify that Node Manager is running on the new Access Server WLS_OAM3.
2. Start the Managed Server using the Administration Console. See the Start the Managed Server
3. Inform WebGates about the new Managed Server. See Inform WebGates of the New Managed Server

- Start the Managed Server
- Inform WebGates of the New Managed Server

Start the Managed Server

To start the Managed Server using the Administration Console:

1. Change to the directory to OAM Domain HOME. For example, `DOMAIN_HOME/bin`
2. Start the Managed Server. For example, enter:

   `./startManagedWebLogic.sh WLS_OAM3 http://hostname:7001`
3. At the prompt, enter the WebLogic username and password. Click **Enter**.
4. Verify that the Managed Server is running. Check the startManagedWebLogic logs, or click **Servers** under **Environment** in the Administration Console to view the Summary page. Refresh the page to see updates.

Inform WebGates of the New Managed Server

To inform any WebGates about the new Managed Server:

1. Log in to the Oracle Access Management Console at `http://OAMHOST1.example.com:7001/oamconsole` as the `oamadmin` user.
2. Click **Application Security** tab, click **Agents** to open SSO Agents page.
3. On the SSO Agents page, click **Search**.
4. Click **Search**.
5. Click the WebGate you want to change.
6. Add the new server to either the primary or secondary server list by clicking the Add **+** icon.
7. Select the server name from the list.
8. Click **Apply**

> **Note:**
>
> Repeat this procedure to inform all the configured WebGate Agents.

Scaling Out Access Manager

You *scale out* to add a new Access Manager managed server to a new node. Scale out is very similar to scale up, but requires the software to be installed on the new node.

1. Install Oracle WebLogic Server on the new host. See Installing Oracle WebLogic Server.
2. Install Identity Management components on the new host. See Installing and Configuring the Access Manager Application Tier.
3. Log in to the Administration Console at `http://`*hostname*`.example.com:7001/oamconsole`.
4. From the Domain Structure window of the Administration Console, expand the **Environment** node and then **Machines**.
5. From the Machines table, click **New**.
6. At the Create a New Machine screen labeled Machine Identity, enter the following information:
   - **Name**: New node host name, for example, *host*.*example*`.com`
   - **Machine OS:** Select operating system, for example, UNIX
7. Click **Next**.
8. In the Create New Machine screen labeled **Node Manager Properties,** enter this information
   - **Type**: Keep the default SSL
   - **Listen Address**: Replace localhost with the hostname that `WLS_OAM3` will run on.

- **Port**: Verify that the Listen Port matches the Node Manager port that will run on the other node, for example, `WLS_OAM3`.

9. Click **Finish**.
10. From the Domain Structure expand Servers.
11. Select a server on the host you want to extend, for example: WLS_OAM1.
12. Click Clone.
13. From the Clone a Server screen labeled Server Identity enter the following:
    - **Server Name**: New name for the server, for example `WLS_OAM3`.
    - **Server Listen Address**: Name of the host the Managed Server will run on.
    - **Server Listen Port**: Port the new managed server will use. This port must be unique within the host.
14. Click **OK**.
15. From the Servers table, click the new clone you just created, for example `WLS_OAM3`.
16. From the Machine option, assign the server to the new machine name you just created. This is the machine that the Managed Server will run on.
17. Click **Save**.
18. Click on the **SSL** tab.
19. Click **Advanced**.
20. Set **Hostname Verification** to **None**.
21. Click **Save**.
22. Run `pack.sh` and `unpack.sh` scripts located at `ORACLE_HOME/oracle_common/common/bin` to pack the domain on OAMHOST1 and unpack it on the new host respectively.

```
pack.sh -domain=ORACLE_HOME/user_projects/domains/domainName -
template =/tmp/idm_domain.jar -template_name="OAM Domain"
unpack.sh -domain=ORACLE_HOME/user_projects/domains/domainName -
template =/tmp/idm_domain.jar
```

- Registering the Managed Server with OAM
- Configuring WebGate with the New OAM Access Server

Registering the Managed Server with OAM

To register the new managed server as an OAM server:

1. Log in to the Oracle Access Management Console at `http://OAMHOST1.example.com:7001/oamconsole` as the `oamadmin` user.
2. Click the **Configuration** tab. Click **Server Instances**.
3. Select **Create** from the Actions menu.
4. Enter the following information:

- **Server Name**: Enter the same server name you entered while cloning the OAM server node in the WebLogic Console.
- **Host**: Host that the server will run on, `OAMHOST3`.
- **Port**: Listen port that was assigned when you created the managed server.
- **OAM Proxy Port**: Port you want the OAM proxy to run on. This is unique for the host.
- **Proxy Server ID:** `AccessServerConfigProxy`
- **Mode**: Select the appropriate mode: `Open`, `Simple`, or `Cert`.

5. Click **Apply**.
6. Edit `oam-config.xml` available at `<DOMAIN_HOME>/config/fmwconfig` to set the IP range of nodes that get added dynamically.

   The Settings Map for the following example is `SetWellKnownAddress>AuthorizedSubnets >Range1 >From [value of start ip range]]>To [value of end ip range].`

```
<Setting Name="AuthorizedSubnets" Type="htf:map">
<Setting Name="Range1" Type="htf:map">
<Setting Name="From" Type="htf:map">
<Setting Name="Key"
Type="xsd:string">oam.coherence.auth.range.from.1</Setting>
<Setting Name="Value"
Type="xsd:string">10.229.139.20</Setting>
</Setting>
<Setting Name="To" Type="htf:map">
<Setting Name="Key"
Type="xsd:string">oam.coherence.auth.range.to.1</Setting>
<Setting Name="Value"
Type="xsd:string">10.229.139.40</Setting>
</Setting>
</Setting>
</Setting>
```

7. Ensure that the `OAM_ORACLE_HOME` property `<ORACLE_HOME>/oam` is set while starting server nodes (OAM1, OAM2 etc). In this environment, `startWeblogic` script is edited to pass `-DOAM_ORACLE_HOME=<ORACLE_HOME>/oam` while starting the Java process.

Configuring WebGate with the New OAM Access Server

Start the Access Server. To use the server, you must inform any WebGates of its existence:

1. Log in to the Oracle Access Management Console at `http://OAMHOST1.example.com:7001/oamconsole` as the `oamadmin` user.
2. Click **Application Security** tab.
3. Click **Agents** to open SSO Agents page
4. On the SSO Agents page, click **Search**.
5. Click the WebGate you want to change.
6. Under the Server Lists section, add the new OAM Access Server `WLS_OAM3` to either the primary or secondary server list by clicking the Add [**+**] icon.

7. Select the server name from the list.
8. Click **Apply**.

Verifying the WebGate Configuration is Updated

To verify the WebGate configuration

1. Log into the Web server where the WebGate was updated previously.
2. Go to the directory `OHSDomain/config/fmwconfig/components/OHS/<instancename>/webgate/config`
3. Open `ObAccessClient.xml` with a text editor. Verify that `primary_server_list` or `secondary_server_list` shows that the new OAM Access Server is updated.

> **Note:**
>
> If the WebGate configuration does not update, recycle the web server, which pulls Webgate Agent profile updates to the `ObAccessClient.xml` file.

Editing Oracle HTTP Server Configuration File

Now that you created and started the new Managed Server, the web tier starts to direct requests to it. However, Oracle recommends informing the web server about the new Managed Server.

In the Web tier, there are several configuration files including `admin_vh.conf`, `sso_vh.conf` and `igdinternal_vh.conf` reside in the directory: *`ORACLE_INSTANCE`*`/config/OHS/`*`component name`*`/moduleconf`. Each contain a number of entries in location blocks. If a block references two server instances and you add a third one, you must update that block with the new server.

Add the new server to the `WebLogicCluster` directive in the file. For example, change:

```
<Location /oam>
   SetHandler weblogic-handler
   WebLogicCluster OAMHOST1.example.com:14100,OAMHOST2.example.com:14100
</Location>
```

to:

```
<Location /oam>
   SetHandler weblogic-handler
   WebLogicCluster
OAMHOST1.example.com:14100,OAMHOST2.example.com:14100,OAMHOST1.example.com:14101
</Location>
```

12

Configuring High Availability for Oracle Directory Services Components

This chapter describes configuring Oracle Directory Services products for high availability in an active-active configuration.

- About the 12c Oracle Directory Services Products
  The following table summarizes Oracle Identity Management products that you can install using the suite-level installation program for 12*c*.
- Configuring Oracle Directory Integration Platform on ODIPHOST1 (OID)
- Prerequisites for Oracle Directory Services High Availability Configuration
  This section describes the prerequisite steps that you must complete before setting up an Oracle Directory Services high availability configuration.
- Oracle Internet Directory High Availability
  This section provides an introduction to Oracle Internet Directory and describes how to design and deploy a high availability environment for Oracle Internet Directory.
- Oracle Directory Integration Platform High Availability
  This section describes how to design and deploy a high availability environment for Oracle Directory Integration Platform (ODIP).
- About Starting and Stopping Oracle Directory Services Components
- About Configuring Oracle Internet Directory for Maximum High Availability
  This section provides high-level instructions for setting up a maximum high availability deployment for Oracle Internet Directory. This deployment includes two sites in different geographic locations. This is an active-active deployment where both sites are active at the same time when the deployment is functioning normally. If one site fails, the surviving site continues to function.

About the 12*c* Oracle Directory Services Products

The following table summarizes Oracle Identity Management products that you can install using the suite-level installation program for 12*c*.

Table 12-1 The 12c Identity Management Components and Product Suites

Product	Description	Product Suite
Oracle Internet Directory	LDAP Version 3-enabled service that enables fast retrieval and centralized management of information about dispersed users, network configuration, and other resources.	Oracle Identity Management Platform and Directory Services Suite

Table 12-1 (Cont.) The 12c Identity Management Components and Product Suites

Product	Description	Product Suite
Oracle Directory Integration Platform	Oracle Directory Integration Platform is a J2EE application that enables you to synchronize data between various directories and the back-end directory. Oracle Directory Integration Platform includes services and interfaces that enable you to deploy synchronization solutions with other enterprise repositories.	Oracle Identity Management Platform and Directory Services Suite
Oracle Directory Services Manager	GUI for Oracle Internet Directory. Oracle Directory Services Manager that simplifies administration and configuration of Oracle Virtual Directory and Oracle Internet Directory by enabling you to use web-based forms and templates. Oracle Directory Services Manager is available from either the Oracle Enterprise Manager Fusion Middleware Control or from its own URL.	Oracle Identity Management Platform and Directory Services Suite

For more information on Oracle Internet Directory installation, See Preparing to Install in *Oracle Fusion Middleware Installing and Configuring Oracle Internet Directory*

Configuring Oracle Directory Integration Platform on ODIPHOST1 (OID)

To configure Oracle Directory Integration Platform on `ODIPHOST1`:

1. Start the Configuration Wizard by running the `ORACLE_HOME/oracle_common/common/bin/config.sh` script (on UNIX) or `ORACLE_HOME\oracle_common\common\bin\config.cmd` (on Windows).

   The **Configuration Type** screen is displayed.

2. Select **Update an existing domain**, and click **Next**.

   The **Templates** screen is displayed.

3. On the **Templates** screen, select **Update Domain Using Product Templates** and then select **Oracle Directory Integration Platform - 12.2.1.3.0[dip]** domain configuration option.

> **Note:**
>
> When you select the **Oracle Directory Integration Platform - 12.2.1.3.0 [dip]** option, **Oracle Enterprise Manager 12.2.1.3.0 [em]** is automatically selected.

Click **Next**.

The **JDBC Data Sources** screen is displayed.

4. Make changes if required and then click **Next**

   The **JDBC Data Sources Test** screen is displayed.

5. Select the data sources to test, and click **Test Selected Connections**.

   Click **Next**.

   The **Database Configuration Type** screen is displayed.

6. Make changes if required and then click **Get RCU Configuration** to retrieve the schema information. After successfully retrieving the schema information, click **Next** to continue.

   The **JDBC Component Schema** screen is displayed.

7. Verify that the values populated are correct for all schemas, and Click **Next**.

> **Note:**
>
> To convert one or more of the schemas to Oracle RAC multi-data source schemas, select the check boxes next to the name of those schemas, and select the **Convert to RAC multi data source** option. Click Next when done. When you click Next, the **Oracle RAC Multi Data Source Component Schema** screen appears.
>
> See Oracle RAC Multi Data Source Component Schema in *Oracle Fusion Middleware Creating WebLogic Domains Using the Configuration Wizard*.

The **JDBC Component Schema Test** screen is displayed.

8. You can select the component schema to test, and click **Test Selected Connections**. Wait for one or more connection tests to complete. If you do not want to test connections, deselect all data sources.

> **Note:**
>
> In order to test connections, the database to which you are trying to connect must be running.

Click **Next**.

The **Advanced Configuration** screen is displayed.

9. Select **Managed Servers, Clusters, and Coherence** option. Click **Next**.

   The **Managed Servers** screen is displayed.

10. Click **Add**, and create one Managed Servers each for `ODIPHOST1` and `ODIPHOST2`.

Table 12-2 Managed Server on ODIPHOST1

Name	Listen Address	Listen Port
`wls_ods1`	`odipHost1.example.com`	7005

Table 12-3 Managed Server on ODIPHOST2

Name	Listen Address	Listen Port
`wls_ods2`	`odipHost2.example.com`	7005

Click **Next**.

The **Clusters** screen is displayed.

11. Click **Add** and enter `odip_cluster` in the **Cluster Name** field to configure cluster for the Managed Servers on `ODIPHOST1` and `ODIPHOST2`.

Click **Next**.

The **Server Templates** screen is displayed.

12. Click **Next** and **Dynamic Servers** screen is displayed.

Click **Next**.

The **Assign Servers to Clusters** screen is displayed.

13. Use the Assign Servers to Clusters screen to assign the `wls_ods1` and `wls_ods2` Managed Servers to the `odip_cluster` cluster. Only Managed Servers appear in the Server list box. The Administration Server is not listed because it cannot be assigned to a cluster.

Select the name of the Managed Server in the **Servers** list box and click the right arrow. The name of the Managed Server is removed from the **Servers** list box and added below the name of the target cluster in the **Clusters** list box.

The name of the Managed Server is removed from the Servers list box and added below the name of the target cluster in the Clusters list box.

Click **Next** and continue clicking **Next** till the **Machines** screen is displayed.

14. Click the **Machine** (for Windows) or **Unix Machine** tab (for UNIX) and then click **Add** to add the following machines:

Table 12-4 Machines

Name	Node Manager Listen Address	Node Manager Listen Port
`odip_1`	`odipHost1.example.com`	5556
`odip_2`	`odipHost2.example.com`	5556

Click **Next**.

The **Assign Servers to Machines** screen is displayed.

15. Use the **Assign Servers to Machines** to assign the WebLogic Server instances to each of the machines.

    a. In the **Machine** list box, select the `odip_1` machine.

    b. Select the `wls_ods1` instance in the **Server** list box and click the right arrow.

       The name of the `wls_ods1` instance is removed from the **Server** list box and added, below the name of the target machine, in the **Machine** list box.

    c. Repeat above steps to assign `odip_2` machine to the `wls_ods2` Managed Server.

    Select the name of the Managed Server in the **Servers** list box and click the right arrow. The name of the Managed Server is removed from the **Servers** list box and added below the name of the target cluster in the **Clusters** list box.

    The name of the Managed Server is removed from the Servers list box and added below the name of the target cluster in the Clusters list box.

    Click **Next** and continue clicking **Next** till the **Configuration Summary** screen is displayed.

16. Review each item on the **Configuration Summary** screen and verify that the information is correct.

    To make any changes, go back to a screen by clicking the Back button or selecting the screen in the navigation pane. Domain creation does not start until you click **Create**.

    A new WebLogic domain (for example: *base_domain*) is created to support Oracle Directory Integration Platform and Fusion Middleware Control in the `<ORACLE_HOME>\user_projects\domains` directory (on Windows). On UNIX, the domain is created in the `<ORACLE_HOME>/user_projects/domains` directory.

Prerequisites for Oracle Directory Services High Availability Configuration

This section describes the prerequisite steps that you must complete before setting up an Oracle Directory Services high availability configuration.

- Oracle Home Requirement
  The Oracle home for the Identity Management components must be the same across all nodes.
- Database Prerequisites
  Several Oracle Identity Management components require the presence of a supported database and schemas.
- About Installing and Configuring the Database Repository
  Oracle recommends a highly available database to store the metadata repository.
- Configuring the Database for Oracle Fusion Middleware 12c Metadata
  You need to have the network prerequisites for deploying an Oracle Identity Management high availability environment.

Oracle Home Requirement

The Oracle home for the Identity Management components must be the same across all nodes.

```
/u01/app/oracle/product/fmw/idm
```

```
/u01/app/oracle/product/fmw/idm
```

Database Prerequisites

Several Oracle Identity Management components require the presence of a supported database and schemas.

To check if your database is certified or to see all certified databases, see the "Certified Databases" section in the Certification Document: http://www.oracle.com/technology/software/products/ias/files/fusion_certification.html.

To determine the database version, run this query:

```
SQL>select version from sys.product_component_version where product like
'Oracle%'
```

About Installing and Configuring the Database Repository

Oracle recommends a highly available database to store the metadata repository.

For maximum availability, Oracle recommends using an Oracle Real Application Clusters (Oracle RAC) database. Oracle recommends that the database use Oracle Automatic Storage Management for data storage. If you use Oracle ASM, the best practice is to also use Oracle Managed Files.

If you use Oracle ASM, install it in its own Oracle Home and have two disk groups:

- One for the Database files.
- One for the Flash Recovery Area.

Oracle Clusterware

See *Oracle Real Application Clusters Installation Guide for Linux and UNIX*.

Automatic Storage Management

See *Oracle Real Application Clusters Installation Guide for Linux and UNIX*.

When you run the installer, select Configure Automatic Storage Management in the Select Configuration page to create a separate Automatic Storage Management home.

Oracle Real Application Clusters

See *Oracle Real Application Clusters Installation Guide for Linux and UNIX*.

Many Oracle Fusion Middleware components require that schemas are in a database prior to installation. Use the Repository Creation Utility (RCU) to create the component schemas in an existing database. For high availability environments, you must create the schemas and load them into an Oracle RAC database.

Configuring the Database for Oracle Fusion Middleware 12c Metadata

You need to have the network prerequisites for deploying an Oracle Identity Management high availability environment.

Create the Oracle Real Application Clusters database to store Oracle Fusion Middleware 12c metadata with the following characteristics:

- It should be in archive log mode to facilitate backup and recovery.
- Optionally, flashback should be enabled.
- It should be created with the ALT32UTF8 character set.

The value of the static PROCESSES initialization parameter must be 500 or greater for Oracle Internet Directory. This value is checked by the Repository Creation Utility.

To check the value, you can use the SHOW PARAMETER command in SQL*Plus:

```
prompt> sqlplus "sys/password as sysdba"
SQL> SHOW PARAMETER processes
```

One common way to change the parameter value is to use a command similar to the following and then stop and restart the database to make the parameter take effect:

```
prompt> sqlplus "sys/password as sysdba"
SQL> ALTER SYSTEM SET PROCESSES=500 SCOPE=SPFILE;
```

The method that you use to change a parameter's value depends on whether the parameter is static or dynamic, and on whether your database uses a parameter file or a server parameter file.

See:

- Database Examples in this Chapter
  See the databases used in Oracle Directory Services Configuration examples in this chapter.
- Configuring Database Services
  Oracle recommends using the Oracle Enterprise Manager Cluster Managed Services Page to create database services that client applications use to connect to the database.
- Verifying Transparent Application Failover
  After the Oracle Internet Directory process starts, you can query the FAILOVER_TYPE, FAILOVER_METHOD, and FAILED_OVER columns in the V$SESSION_VIEW to obtain information about connected clients and their TAF status.
- Configuring Virtual Server Names and Ports for the Load Balancer
  There are network prerequisites for Load Balancer and Virtual Server Names for deploying an Oracle Identity Management high availability environment.

Database Examples in this Chapter

See the databases used in Oracle Directory Services Configuration examples in this chapter.

Table 12-5 Databases Used in Identity Management Configuration Examples

Component	Database Service Name	Database Instance Name
Oracle Internet Directory	oid.example.com	oiddb1, oiddb2
Oracle Directory Integration Platform	oid.example.com	oiddb1, oiddb2
Oracle Directory Services Manager	N/A	N/A

Configuring Database Services

Oracle recommends using the Oracle Enterprise Manager Cluster Managed Services Page to create database services that client applications use to connect to the database.

You can also use SQL*Plus to configure your Oracle RAC database to automate failover for Oracle Internet Directory using the following instructions. Note that each of the following commands has to be run on only one node in the cluster:

1. Use the CREATE_SERVICE subprogram to both create the database service and enable high availability notification and configure server-side Transparent Application Failover (TAF) settings.

   ```
   prompt> sqlplus "sys/password as sysdba"

   SQL> EXECUTE DBMS_SERVICE.CREATE_SERVICE
   (SERVICE_NAME => 'idm.example.com',
   NETWORK_NAME => 'idm.example.com',
   AQ_HA_NOTIFICATIONS => TRUE,
   FAILOVER_METHOD => DBMS_SERVICE.FAILOVER_METHOD_BASIC,
   FAILOVER_TYPE => DBMS_SERVICE.FAILOVER_TYPE_SELECT,
   FAILOVER_RETRIES => 5, FAILOVER_DELAY => 5);
   ```

   You must enter the EXECUTE DBMS_SERVICE command on a single line.

2. Add the service to the database and assign it to the instances using `srvctl`.

   ```
   prompt> srvctl add service -d idmdb -s idm -r idmdb1,idmdb2
   ```

3. Start the service using `srvctl`.

   ```
   prompt> srvctl start service -d idmdb -s  idm
   ```

If you already have a service in the database, ensure that it is enabled for high availability notifications and configured with the proper server-side Transparent Application Failover (TAF) settings. Use the DBMS_SERVICE package to modify the service to enable high availability notification to go through Advanced Queuing (AQ) by setting the AQ_HA_NOTIFICATIONS attribute to TRUE and configure server-side TAF settings, as shown below:

```
prompt> sqlplus "sys/password as sysdba"

SQL> EXECUTE DBMS_SERVICE.MODIFY_SERVICE
(SERVICE_NAME => 'idm.example.com',
AQ_HA_NOTIFICATIONS => TRUE,
FAILOVER_METHOD => DBMS_SERVICE.FAILOVER_METHOD_BASIC,
FAILOVER_TYPE => DBMS_SERVICE.FAILOVER_TYPE_SELECT,
FAILOVER_RETRIES => 5, FAILOVER_DELAY => 5);
```

You must enter the EXECUTE DBMS_SERVICE command on a single line.

See Also:

- Administering Services with Oracle Enterprise Manager, PL/SQL, and SRVCTL in *Oracle Real Application Clusters Administration and Deployment Guide*
- DBMS_SERVICE in *Oracle Database PL/SQL Packages and Types Reference*

Verifying Transparent Application Failover

After the Oracle Internet Directory process starts, you can query the FAILOVER_TYPE, FAILOVER_METHOD, and FAILED_OVER columns in the V$SESSION_VIEW to obtain information about connected clients and their TAF status.

For example, use the following SQL statement to verify that TAF is correctly configured:

```
SELECT MACHINE, FAILOVER_TYPE, FAILOVER_METHOD, FAILED_OVER, COUNT(*)
FROM V$SESSION
GROUP BY MACHINE, FAILOVER_TYPE, FAILOVER_METHOD, FAILED_OVER;
```

The output before failover is similar to this:

```
MACHINE              FAILOVER_TYPE FAILOVER_M  FAI   COUNT(*)
-------------------- ------------- ---------- ---- ----------
oidhost1             SELECT        BASIC      NO           11
oidhost1             SELECT        BASIC      NO            1
```

The output after failover is similar to this:

```
MACHINE              FAILOVER_TYPE FAILOVER_M  FAI   COUNT(*)
-------------------- ------------- ---------- ---- ----------
oidhost2             SELECT        BASIC      NO           11
oidhost2             SELECT        BASIC      NO            1
```

Configuring Virtual Server Names and Ports for the Load Balancer

There are network prerequisites for Load Balancer and Virtual Server Names for deploying an Oracle Identity Management high availability environment.

- Load Balancers
  All components in the Oracle Identity Management software stack require a hardware load balancer when deployed in a high availability configuration.
- Virtual Server Names
  You should setup virtual server names for the high availability deployments. Ensure that the virtual server names are associated with IP addresses and are part of your Domain Name System (DNS). The computers on which Oracle Fusion Middleware is running must be able to resolve these virtual server names.

Load Balancers

All components in the Oracle Identity Management software stack require a hardware load balancer when deployed in a high availability configuration.

The hardware load balancer should have the following features:

- Ability to load-balance traffic to a pool of real servers through a virtual hostname: Clients access services using the virtual hostname (instead of using actual host names). The load balancer can then load balance requests to the servers in the pool.
- Port translation configuration: The load balancer should have the ability to perform port translation, where it enables incoming requests received on one port to be routed to a server process running on a different port. For example, a request received on port 80 can be routed to port 7777.
- Protocol translation: The load balancer should support protocol translation between systems running different protocols. It enables users on one network to access hosts on another network, despite differences in the native protocol stacks associated with the originating device and the targeted host. For example, incoming requests can be HTTPS, and outgoing requests can be HTTP.

  This feature is recommended but not required.
- SSL acceleration: SSL acceleration is a method of offloading the processor-intensive public key encryption algorithms involved in SSL transactions to a hardware accelerator.

  This feature is recommended but not required.
- Monitoring of ports (HTTP, HTTPS, LDAP, LDAPS)
- Virtual servers and port configuration. Ability to configure virtual server names and ports on your external load balancer, and the virtual server names and ports must meet the following requirements:

  The load balancer should enable configuration of multiple virtual servers. For each virtual server, the load balancer should enable configuration of traffic management on more than one port. For example, for Oracle Internet Directory clusters, the load balancer needs to be configured with a virtual server 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 load balancer through the virtual server names.
- Ability to detect node failures and immediately stop routing traffic to the failed node.
- Resource monitoring / port monitoring / process failure detection.

  The load balancer must be able 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 load balancer has the ability to automatically detect failures, you should use it.
- Fault-tolerant mode

  It is highly recommended that you configure the load balancer to be in fault-tolerant mode.
- Other

  Oracle recommends that you configure the load balancer virtual server to return immediately to the calling client when the back-end services that it forwards traffic to 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.
- Sticky routing capability

Ability to maintain sticky connections to components based on cookies or URL.

The following table shows the virtual server names to use for the external load balancer in the Oracle Identity Management high availability environment.

Table 12-6 Virtual Server Names for the External Load Balancer

Component	Virtual Server Name
Oracle Internet Directory	oid.example.com
Oracle Directory Services Manager Console	admin.example.com

Virtual Server Names

You should setup virtual server names for the high availability deployments. Ensure that the virtual server names are associated with IP addresses and are part of your Domain Name System (DNS). The computers on which Oracle Fusion Middleware is running must be able to resolve these virtual server names.

oid.example.com

This virtual server acts as the access point for all LDAP traffic to the Oracle Internet Directory servers in the directory tier. Traffic to both the SSL and non-SSL ports is configured. The clients access this service using the address oid.example.com:636 for SSL and oid.example.com:389 for non-SSL.

Monitor the heartbeat of the Oracle Internet Directory processes on OIDHOST1 and OIDHOST2. If an Oracle Internet Directory process stops on OIDHOST1 or OIDHOST2, or if either host is down, the load balancer must continue to route the LDAP traffic to the surviving computer.

Oracle Internet Directory High Availability

This section provides an introduction to Oracle Internet Directory and describes how to design and deploy a high availability environment for Oracle Internet Directory.

- About Oracle Internet Directory Component Architecture
  Oracle Internet Directory is an LDAP store that can be used by Oracle components such as Directory Integration Platform, Oracle Directory Services Manager, JPS, and also by non-Oracle components. These components connect to Oracle Internet Directory using the LDAP or LDAPS protocols.
- Understanding Oracle Internet Directory High Availability Concepts
  This section provides conceptual information about using Oracle Internet Directory in a high availability two-node Cluster Configuration.
- Oracle Internet Directory High Availability Configuration Steps
  You can deploy Oracle Internet Directory in a High Availability configuration as part of a WebLogic Server domain.
- Validating Oracle Internet Directory High Availability
  Use the `ldapbind` command-line tool to ensure that you can connect to each OID instance and the LDAP Virtual Server. The `ldapbind` tool enables you to determine whether you can authenticate a client to a server.

- Oracle Internet Directory Failover and Expected Behavior
  This section describes how to perform a failover of Oracle Internet Directory and Oracle RAC.
- Troubleshooting Oracle Internet Directory High Availability
  This section provides information that can help you troubleshoot OID high availability issues:
- Additional Oracle Internet Directory High Availability Issues
  This section describes issues for Oracle Internet Directory in a high availability environment.

About Oracle Internet Directory Component Architecture

Oracle Internet Directory is an LDAP store that can be used by Oracle components such as Directory Integration Platform, Oracle Directory Services Manager, JPS, and also by non-Oracle components. These components connect to Oracle Internet Directory using the LDAP or LDAPS protocols.

The Oracle directory replication server uses LDAP to communicate with an Oracle directory (LDAP) server instance. To communicate with the database, all components use OCI/Oracle Net Services. Oracle Directory Services Manager and the command-line tools communicate with the Oracle directory servers over LDAP.

An Oracle Internet Directory node consists of one or more directory server instances connected to the same directory store. The directory store—that is, the repository of the directory data—is an Oracle database.

An Oracle Internet Directory node includes the following major elements:

Table 12-7 An Oracle internet Directory Node

Element	Description
Oracle directory server instance	Also called either an LDAP server instance or a directory server instance, it services directory requests through a single Oracle Internet Directory dispatcher process listening at specific TCP/IP ports. There can be more than one directory server instance on a node, listening on different ports.
Oracle directory replication server	Also called a replication server, it tracks and sends changes to replication servers in another Oracle Internet Directory system. There can be only one replication server on a node. You can choose whether to configure the replication server. If there are multiple instances of Oracle Internet Directory that use the same database, only one of them can be running replication. This is true even if the Oracle Internet Directory instances are on different nodes. The replication sever process is a process within Oracle Internet Directory. It only runs when replication is configured. For more information on Oracle Internet Directory replication, see Configuring Identity Management for Maximum High Availability..

Table 12-7 (Cont.) An Oracle internet Directory Node

Element	Description
Oracle Database Server	Stores the directory data. Oracle strongly recommends that you dedicate a database for use by the directory. The database can reside on the same node as the directory server instances.
OID Monitor (OIDMON)	Initiates, monitors, and terminates the LDAP server and replication server processes. When you invoke process management commands, such as oidctl or Node Manager, or when you use Fusion Middleware Control to start or stop server instances, your commands are interpreted by this process. OIDMON also monitors servers and restarts them if they have stopped running for abnormal reasons. OIDMON starts a default instance of OIDLDAPD. If the default instance of OIDLDAPD is stopped using the OIDCTL command, then OIDMON stops the instance. When OIDMON is restarted by Node Manager (using `startComponent.sh`), OIDMON restarts the default instance. All OID Monitor activity is logged in the file `DOMAIN_HOME/servers/OID/logs/oid1/oidmon-xxxx.log`. This file is on the Oracle Internet Directory server file system.
OID Control Utility (OIDCTL)	Communicates with OID Monitor by placing message data in Oracle Internet Directory server tables. This message data includes configuration parameters required to run each Oracle directory server instance. Normally used from the command line only to stop and start the replication server.

- Oracle Internet Directory Component Characteristics
  Oracle Internet Directory, which is Oracle's LDAP store, is a C-based component that uses a database as its persistence store. It is a stateless process and stores all of the data and the majority of its configuration information in the back-end database. It uses Oracle Net Services to connect to the database.

Oracle Internet Directory Component Characteristics

Oracle Internet Directory, which is Oracle's LDAP store, is a C-based component that uses a database as its persistence store. It is a stateless process and stores all of the data and the majority of its configuration information in the back-end database. It uses Oracle Net Services to connect to the database.

- Runtime Processes
  Oracle Internet Directory has the following runtime processes:

- Process Lifecycle
  Node Manager is responsible for the direct start, stop, restart and monitoring of the daemon process, OIDMON (ORACLE_HOME/bin/oidmon). OIDMON is responsible for the process control of an Oracle Internet Directory instance.
- Request Flow
  Once the Oracle Internet Directory (OID) process starts up, clients access OID using the LDAP or LDAPS protocol. There is no affect on other running instances when an OID instance starts up
- About Configuration Artifacts
  The storage location requires a DB connect string. `TNSNAMES.ORA` is stored in DOMAIN_HOME/config. The wallet is stored in `DOMAIN_HOME/config/fmwconfig/components/OID/admin` (The DB ODS user password is stored in the wallet).
- External Dependencies
  Oracle Internet Directory uses an Oracle database to store configuration information as well as data. It uses the ODS schema to store this information.
- Oracle Internet Directory Log File
  Log files for Oracle Internet Directory are under the following directory:

Runtime Processes

Oracle Internet Directory has the following runtime processes:

- **OIDLDAPD**: This is the main process for Oracle Internet Directory. OIDLDAPD consists of a dispatcher process and a server process. The dispatcher process spawns the OIDLDAPD server processes during startup. Each OIDLDAPD dispatcher process has its own SSL and non-SSL ports for receiving requests. Every OID instance has one dispatcher and one server process by default. The number of server processes spawned for an instance is controlled by the `orclserverprocs` attribute.
- **OIDMON**: OIDMON is responsible for the process control of an Oracle Internet Directory instance. This process starts, stops, and monitors Oracle Internet Directory. During startup OIDMON spawns the OIDLDAPD dispatcher process and the replication server process, if replication is configured for the instance.
- **Replication server process**: This is a process within Oracle Internet Directory that runs only when replication is configured. The replication server process is spawned by OIDMON during startup.
- **Node Manager**: Node Manager is a daemon process that monitors Oracle Fusion Middleware components, including Oracle Internet Directory.

  Node Manager is responsible for the direct start, stop, restart and monitoring of OIDMON. It does not start or stop the server process directly.

Process Lifecycle

Node Manager is responsible for the direct start, stop, restart and monitoring of the daemon process, OIDMON (ORACLE_HOME/bin/oidmon). OIDMON is responsible for the process control of an Oracle Internet Directory instance.

Process Status Table

Oracle Internet Directory process information is maintained in the ODS_PROCESS_STATUS table in the ODS database user schema. OIDMON reads the contents of the table at a specified interval and acts upon the intent conveyed by the contents of that table. The interval is controlled by the value of the sleep command line argument used at OIDMON startup, and the default value is 10 seconds.

Starting and Stopping Oracle Internet Directory

An Oracle Internet Directory instance can be started and stopped using system component management scripts — `startComponent.sh` and `stopComponent.sh`.

Start Process

The start process for Oracle Internet Directory is:

1. Upon receiving the start command, Node Manager issues an oidmon start command with appropriate arguments.
2. OIDMON then starts all Oracle Internet Directory Server instances whose information in the ODS_PROCESS_STATUS table has state value 1 or 4 and COMPONENT_NAME, INSTANCE_NAME values matching the environment parameters set by Node Manager.

Stop Process

The stop process for Oracle Internet Directory is:

1. Upon receiving the stop command, Node Manager issues an oidmon stop command.
2. For each row in the ODS_PROCESS_STATUS table that matches the environment parameters COMPONENT_NAME, and INSTANCE_NAME, the oidmon stop command kills OIDMON, OIDLDAPD, and OIDREPLD processes and updates the state to 4.

Monitoring

Node Manager does not monitor server processes directly. Node Manager monitors OIDMON and OIDMON monitors the server processes. The events are:

- When you start OIDMON through Node Manager, Node Manager starts OIDMON and ensures that OIDMON is up and running.
- If OIDMON goes down for some reason, Node Manager brings it back up.
- OIDMON monitors the status of the Oracle Internet Directory dispatcher process, LDAP server processes, and replication server process and makes this status available to Node Manager.

Request Flow

Once the Oracle Internet Directory (OID) process starts up, clients access OID using the LDAP or LDAPS protocol. There is no affect on other running instances when an OID instance starts up

Oracle Internet Directory listener/dispatcher starts a configured number of server processes at startup time. The number of server processes is controlled by the orclserverprocs attribute in the instance-specific configuration entry. The default value for orclserverprocs is 1. Multiple server processes enable OID to take advantage of multiple processor systems.

The OID dispatcher process sends the LDAP connections to the OID server process in a round robin fashion. The maximum number of LDAP connections accepted by each server is 1024 by default. This number can be increased by changing the attribute orclmaxldapconns in the instance-specific configuration entry, which has a DN of the form:

```
cn=componentname,cn=osdldapd,cn=subconfigsubentry
```

Database connections from each server process are spawned at server startup time, depending on the value set for the instance configuration parameters ORCLMAXCC and ORCLPLUGINWORKERS. The number of database connections spawned by each server equals ORCLMAXCC + ORCLPLUGINWORKERS + 2. The OID server processes communicate with the Oracle database server through Oracle Net Services. An Oracle Net Services listener/dispatcher relays the request to the Oracle database. For more information, see **Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory**.

About Configuration Artifacts

The storage location requires a DB connect string. `TNSNAMES.ORA` is stored in DOMAIN_HOME/config. The wallet is stored in `DOMAIN_HOME/config/fmwconfig/components/OID/admin` (The DB ODS user password is stored in the wallet).

External Dependencies

Oracle Internet Directory uses an Oracle database to store configuration information as well as data. It uses the ODS schema to store this information.

The Oracle directory replication server uses LDAP to communicate with an Oracle directory (LDAP) server instance. To communicate with the database, all components use OCI/Oracle Net Services. Oracle Directory Services Manager and the command-line tools communicate with the Oracle directory servers over LDAP.

Oracle Internet Directory Log File

Log files for Oracle Internet Directory are under the following directory:

`DOMAIN_HOME/servers/OID/logs/`*`InstanceName`*`/`

Table shows Oracle Internet Directory processes and the log file name and location for the process.

Table 12-8 Locations of Oracle Internet Directory Process Log Files

Process	Log File Location
Directory server (oidldapd)	`DOMAIN_HOME/servers/OID/logs/InstanceName/oidldapd00sPID-XXXX.log` where: 00 is the instance number (00 by default) s stands for server PID is the server process identifier XXXX is a number from 0000 to orclmaxlogfilesconfigured. Once the orclmaxlogfilesconfigured value is reached, it starts over again from 0000. When it starts over, it truncates the file to 0 bytes. `DOMAIN_HOME/servers/OID/logs/InstanceName/oidstackInstNumberPID.log`
LDAP dispatcher (oiddispd)	`DOMAIN_HOME/servers/OID/logs/InstanceName/oiddispd00-XXXX.log` where: 00 is the instance number (00 by default) XXXX is a number from 0000 to orclmaxlogfilesconfigured
OID Monitor (OIDMON)	`DOMAIN_HOME/servers/OID/logs/InstanceName/oidmon-XXXX.log` where: XXXX is a number from 0000 to orclmaxlogfilesconfigured
Directory replication server (oidrepld)	`DOMAIN_HOME/servers/OID/logs/InstanceName/oidrepld-XXXX.log` where: XXXX is a number from 0000 to orclmaxlogfilesconfigured

For more information on using log files to troubleshoot Oracle Internet Directory, see Troubleshooting Oracle Internet Directory High Availability.

Understanding Oracle Internet Directory High Availability Concepts

This section provides conceptual information about using Oracle Internet Directory in a high availability two-node Cluster Configuration.

See Oracle Internet Directory Prerequisites for prerequisites and Oracle Internet Directory High Availability Configuration Steps to set up the two-node Cluster Configuration.

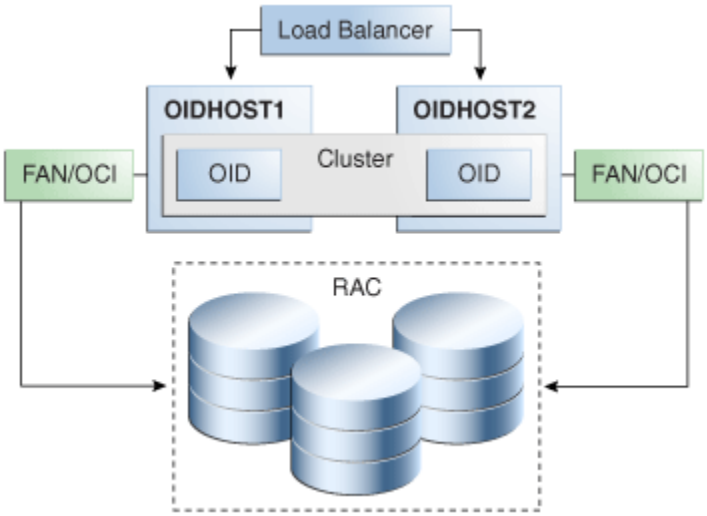
- Oracle Internet Directory High Availability Architecture
  Learn about the Oracle Internet Directory Cluster Configuration high availability architecture in an active-active configuration.
- Protection from Failures and Expected Behavior
  This section discusses protection from different types of failure in an OID Cluster Configuration.
- Oracle Internet Directory Prerequisites
  This section describes prerequisites for setting up the OID high availability architecture.

Oracle Internet Directory High Availability Architecture

Learn about the Oracle Internet Directory Cluster Configuration high availability architecture in an active-active configuration.

The Figure 12-1 shows the Oracle Internet Directory Cluster Configuration high availability architecture in an active-active configuration.

Figure 12-1 Oracle Internet Directory Cluster Configuration High Availability Architecture



The Figure 12-1 shows Oracle Internet Directory (OID) in the directory tier in a Cluster Configuration high availability architecture. Clustering is set up at installation time. The load balancing router routes LDAP client requests to the two OID instances that are clustered on `OIDHOST1`and `OIDHOST2`.

Transparent Application Failover (TAF) is used to connect the OID instances with the Oracle RAC database that serves as the security metadata repository. The Oracle RAC database is configured in `TNSNAMES.ORA`. High availability event notification is used for notification when an Oracle RAC instance becomes unavailable.

- Starting and Stopping the Cluster
  In the Cluster Configuration, Node Manager (`startComponent.sh` and `stopComponent.sh` commands) start each OID instance. There is no affect on OID at startup. A new database connection spawns when OID starts.
- Cluster-Wide Configuration Changes (OID)

Starting and Stopping the Cluster

In the Cluster Configuration, Node Manager (`startComponent.sh` and `stopComponent.sh` commands) start each OID instance. There is no affect on OID at startup. A new database connection spawns when OID starts.

When the cluster is stopped using Node Manager (`stopComponent.sh` command), OID disconnects from the database and the OID server stops.

Cluster-Wide Configuration Changes (OID)

When you deploy Oracle Internet Directory in a high availability configuration, all Oracle Internet Directory instances in the cluster share the same database. Any changes made to Oracle Directory Integration Platform on one Oracle Internet Directory node automatically propagate to all the Oracle Internet Directory instances in the cluster.

Directory Synchronization Profiles

Changes that you make to directory integration profiles on one Oracle Internet Directory node do not replicate automatically to other Oracle Internet Directory nodes in a default multimaster Oracle Internet Directory replication environment. You must copy changes from the primary node to the secondary nodes manually and do so on a periodic basis. By doing this, a directory synchronization profile can run on a secondary node if a problem occurs on the primary node.

Oracle Directory Integration Platform uses the parameter `orcllastappliedchangenumber`. The value assigned to the `lastchangenumber` attribute in a directory synchronization profile depends on the directory server on which Oracle Directory Integration Platform is running. In an active-active Oracle Directory Integration Platform configuration, you must manually update the `lastchangenumber` attribute in all instances.

To synchronize directory provisioning profiles between the primary Oracle Internet Directory node and secondary nodes:

1. On the primary node, use the `ldifwrite` command to create an LDIF dump of the entries from this container:

   ```
   cn=subscriber profiles,cn=changelog subscriber,cn=oracle internet directory
   ```

2. Copy the LDIF dump to the secondary node.
3. Use the `ldapadd` command to add the profiles on the secondary node.

After you copy an export profile to a target node, you must update the `lastchangenumber` attribute with the target node value. To update the value:

1. Disable the synchronization profile.
2. Get the value of the `lastchangenumber` attribute on the target node using the `ldapsearch` command.
3. Use `ldapsearch` to get the LDIF dump of the profile entry.
4. Use `ldapadd` to add the profile to the other Managed Server instance.
5. Go to the Oracle Directory Integration Platform Admin console and select the profile. Select **Edit**. Select the **Advanced** tab then select **Edit and Persist**. Enter the value of the `lastchangenumber` attribute. Save the profile.
6. Enable the synchronization profile.

Directory Provisioning Profiles

In a default multimaster Oracle Internet Directory replication environment, Oracle Directory Integration Platform is installed in the same location as the primary Oracle Internet Directory. The information and steps in this topic applies only when multimaster replication is set up.

If the primary node fails, event propagation stops for all profiles located on the node. Although the events are queued and not lost while the primary node is stopped, the events do

not propagate to any applications that expect them. To ensure that events continue to propagate even when the primary node is down for the Version 1.0 and 2.0 profiles, the directory provisioning profiles must be copied to other secondary nodes.

However, copy directory provisioning profiles from the primary node to any secondary nodes immediately after an application is installed and before any user changes are made in Oracle Internet Directory.

To synchronize directory provisioning profiles between a primary node and any secondary nodes:

1. On the primary node, use the ldifwrite command to create an LDIF dump of the entries from this container:

   ```
   cn=provisioning profiles,cn=changelog subscriber,cn=oracle internet directory
   ```

2. Copy the LDIF dump to the secondary node.
3. Use the `ldapadd` command to add the profiles on the secondary node.

Protection from Failures and Expected Behavior

This section discusses protection from different types of failure in an OID Cluster Configuration.

- Oracle Internet Directory Process Failure
  OIDMON monitors OID processes. If the OID process goes down, OIDMON tries to restart it.
- Expected Client Application Behavior When Failure Occurs
  Oracle Internet Directory server failure is usually transparent to OID clients as they continue to get routed through the load balancer. External load balancers are typically configured to perform a health check of OID processes. If a request is received before the load balancer detects process unavailability, clients application could receive a error. If the client application performs a retry, the load balancer should route it to a healthy OID instance and the request should be successful.
- External Dependency Failure
  This section describes the protection available for OID from database failure.

Oracle Internet Directory Process Failure

OIDMON monitors OID processes. If the OID process goes down, OIDMON tries to restart it.

Node Manager monitors OIDMON. If OIDMON goes down, Node Manager restarts OIDMON.

If you cannot start an OID process, the front-ending load balancing router detects failure of OID instances in the Cluster Configuration and routes LDAP traffic to surviving instances. In case of failure, the LDAP client retries the transaction. If the instance fails in the middle of a transaction, the transaction is not committed to the database. When the failed instance comes up again, the load balancing router detects this and routes requests to all the instances.

If an OID instance in the Cluster Configuration gets hung, the load balancing router detects this and routes requests to surviving instances.

If one OID instance in the two-node Cluster Configuration fails (or if one of the computers hosting an instance fails), the load balancing router routes clients to the surviving OID instance.

Expected Client Application Behavior When Failure Occurs

Oracle Internet Directory server failure is usually transparent to OID clients as they continue to get routed through the load balancer. External load balancers are typically configured to perform a health check of OID processes. If a request is received before the load balancer detects process unavailability, clients application could receive a error. If the client application performs a retry, the load balancer should route it to a healthy OID instance and the request should be successful.

In OID active-active configurations, if you are doing ldapadd operations through the LDIF file at the time of failover, your operation would fail even if you are doing this operation through a load balancer host and port. This is because OID is down for a fraction of a second. For most applications, this will not be an issue because most applications have the ability to retry the connection a fixed number of times.

External Dependency Failure

This section describes the protection available for OID from database failure.

By default, the `tnsnames.ora` file configured in OID's ORACLE_INSTANCE ensures that OID's connections to the database are load balanced between the Oracle RAC database instances. For example, if an OID instance establishes four database connections, two connections are made to each database instance.

Oracle Internet Directory uses database high availability event notification to detect database node failure and to fail over to a surviving node.

If Transparent Application Failover (TAF) is configured, then upon a database instance failure, OID will fail over its database connections to the surviving database instance, which enables the LDAP search operations that were in progress during the failover to be continued.

If both TAF and high availability event notification are configured, TAF is used for failover and high availability event notifications are used only for logging the events. The high availability event notifications are logged in OIDLDAPD log file.

Oracle Internet Directory also has a mechanism to detect stale database connections, which enables OID to reconnect to the database.

If none of the database instances are available for a prolonged period, then the OID LDAP and REPL processes will automatically be shut down. However, OIDMON and Node Manager will continue to ping for the database instance availability and when the database becomes available, the OID processes (LDAP and REPL) are automatically restarted by OIDMON.

While all database instances are down, OIDMON continues to be up and an `oid_instanceStatus(instanceName = 'instance-name')` command shows that OIDLDAPD instances are down. When a database instance becomes available, OIDMON restarts all configured OID instances.

All database failover induced activity for OID is recorded in the OIDMON log file.

Oracle Internet Directory Prerequisites

This section describes prerequisites for setting up the OID high availability architecture.

- Synchronizing the Time on Oracle Internet Directory Nodes
  Before setting up OID in a high availability environment, you must ensure that the time on the individual OID nodes is synchronized.
- Load Balancer Virtual Server Names for Oracle Internet Directory
  When you deploy OID in a high availability configuration, Oracle recommends using an external load balancer to front-end OID instances and load balance requests between the OID instances.

Synchronizing the Time on Oracle Internet Directory Nodes

Before setting up OID in a high availability environment, you must ensure that the time on the individual OID nodes is synchronized.

Synchronize the time on all nodes 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 OID servers on its node.

If there are more than two nodes, the same behavior is followed. For example, assume that there are three nodes, where the first node is 150 seconds ahead of the second node, and the second node is 150 seconds ahead of the third node. In this case, the third node is 300 seconds behind the first node, so the OID Monitor will not start the servers on the third node until the time is synchronized.

Load Balancer Virtual Server Names for Oracle Internet Directory

When you deploy OID in a high availability configuration, Oracle recommends using an external load balancer to front-end OID instances and load balance requests between the OID instances.

See Configuring Virtual Server Names and Ports for the Load Balancer.

Oracle Internet Directory High Availability Configuration Steps

You can deploy Oracle Internet Directory in a High Availability configuration as part of a WebLogic Server domain.

Oracle recommends that you set up OID in a clustered deployment in which clustered OID instances access the same Oracle RAC database repository.

- Installing Oracle Fusion Middleware Components
  This section describes how to install the required binaries for the Oracle WebLogic Server (WL_HOME) and Oracle Home for (ORACLE_HOME) for Oracle Identity Management.
- Creating Oracle Internet Directory Schemas in the Repository Using RCU
  This section describes the procedure to create schemas in the repository using Repository Creation Utility (RCU).
- Configuring Oracle Internet Directory With a WebLogic Domain
  In this configuration, OID and a WebLogic Server domain is configured on the first host and the second host. The OID instance on the second host joins the domain created on the first host.

Installing Oracle Fusion Middleware Components

This section describes how to install the required binaries for the Oracle WebLogic Server (WL_HOME) and Oracle Home for (ORACLE_HOME) for Oracle Identity Management.

Oracle strongly recommends that you read the release notes for any additional installation and deployment considerations prior to starting the setup process.

- Installing Oracle WebLogic Server
  This section describes the procedure to install Oracle WebLogic server.
- Installing Oracle Internet Directory
  This section describes the procedure to install Oracle Internet Directory.

Installing Oracle WebLogic Server

This section describes the procedure to install Oracle WebLogic server.

See Understanding Your Installation Starting Point in *Oracle Fusion Middleware Installation Planning Guide* for the Oracle WebLogic Server version to use with the latest Oracle Fusion Middleware version.

Ensure that system, patch, kernel and other requirements are met as described in Installation Prerequisites in *Oracle Fusion Middleware Installation Guide for Oracle WebLogic Server*.

Start the Oracle WebLogic Server installer then follow these steps:

1. On the Welcome screen, click **Next**.
2. On the Choose Installation Location screen, browse and navigate to the folder where you want to install the WebLogic Servre.

   Click **Next**
3. On the Installation Type screen, Select **Fusion Middleware Insfrastructure**
4. On the Prerequisite Checks screen, Click **Next**.
5. On the Installation Summary screen, the window contains a list of the components you selected for installation, along with the approximate amount of disk space to be used by the selected components once installation is complete.

   Click **Install**.
6. On the Installation Progress, Click **Next**.
7. On the **Installation Complete** screen, click **Finish**.

Installing Oracle Internet Directory

This section describes the procedure to install Oracle Internet Directory.

Ensure that the system, patch, kernel and other requirements are met. These are listed in the Preparing to Install in *Oracle Fusion Middleware Installation Guide for Oracle Identity Management*.

> **Note:**
>
> Ensure that the ORACLE_HOME that are using for installing OID is same as ORACLE_HOME used for installing weblogic server.

On Linux platforms, if the `/etc/oraInst.loc` file exists, verify that its contents are correct. Specifically, check that the inventory directory is correct and that you have write permissions for it. If the`/etc/oraInst.loc` file does not exist, skip this step.

Start the installer for Oracle Fusion Middleware components.

Before starting the install, ensure that the following environment variables are not set:

- LD_ASSUME_KERNEL

On the Specify Inventory Directory screen, do the following:

- Enter `HOME/oraInventory`, where HOME is the home directory of the user performing the installation (this is the recommended location).
- Enter the OS group for the user performing the installation. Click **Next**.

For a UNIX install, follow the instructions on screen to run `createCentralInventory.sh` as root.

Click OK.

Proceed as follows:

1. Start Oracle Internet Directory 12c Installer.
2. On the Welcome screen, click **Next**.
3. On the **Auto Updates** screen, select **Skip Auto Updates** and click **Next**.
4. On the **Installation Location** screen, browse and select the folder where you want to install Oracle Internet Directory. Click **Next**

   > **Note:**
   >
   > Ensure that the ORACLE_HOME used for installing OID is same as ORACLE_HOME used for installing Weblogic server.

5. On the **Installation Type** screen, Based on your requirement, Select either of the option — **Standalone Oracle Internet Directory Server (Managed Independently of WebLogic server)** or **Collocated Oracle Internet Directory Server (Managed through Weblogic server)** . Click**Next**.
6. On the **JDK Selection** screen, browse and select `jdk8` folder and click **Next**.
7. On the **Prerequisite Checks** ensure that all the prerequisites are met, without any warnings. Click **Next**.
8. On the **Installation Summary** screen, click **Install**.
9. Click **Finish**.

Creating Oracle Internet Directory Schemas in the Repository Using RCU

This section describes the procedure to create schemas in the repository using Repository Creation Utility (RCU).

To run RCU and create Identity Management schemas in a RAC database repository:

1. Run this command:

   ```
   ORACLE_HOME/oracle_common/bin/rcu &
   ```

2. On the Welcome screen, click **Next**.
3. On the Create Repository screen, select the **Create Repository** and **System Load and Product Load** to load component schemas into an existing database.

   Click **Next**.
4. On the Database Connection Details screen, enter connection information for the existing database as follows:
   - **Database Type:** `Oracle Database`
   - **Connection String Format**: select either —
     - Connection Parameters: This option provides an interface that accepts all connection parameters (namely - host, port and service name) separately in different UI elements.
     - Connection String: This option accepts all parameters in a single string. This string can be of one of the following formats:

       `<host>:<port>/service` **or** `<host>:<port>:<SID>` **or** `(DESCRIPTION=(ADDRESS=(host=host_name)(protocol=protocol_name)(port=port_number))(CONNECT_DATA=(SERVICE_NAME=service_name)))`
   - **Host Name:**Name of the computer on which the database is running. For an Oracle RAC database, specify the VIP name or one node name. Example: `INFRADBHOST1-VIP or INFRADBHOST2-VIP`
   - **Port:**The port number for the database. Example: `1521`
   - **Service Name:**The service name of the database. Example: `oid.example.com`
   - **Username:**`SYS`
   - **Password**: The SYS user password
   - **Role:**`SYSDBA`
5. Click **Next.**
6. On the **Select Components** screen, create a new prefix and select the components to be associated with this deployment:

   Create a New Prefix:`idm` (Entering a prefix is optional if you select only **Identity Management**(Oracle Internet Directory - ODS) in the **Components** field)

   Components: Select **Identity Management**(Oracle Internet Directory - ODS). On selecting `Identity Management` component, some of the default components that are dependent on Oracle Internet Directory are automatically selected.

   Click **Next**.

7. On the Schema Passwords screen, enter the passwords to create password for the main and auxiliary schema users.

   Click **Next**.

8. On the **Map Tablespaces** screen, select the tablespaces for the components. The default tablespaces for the selected components are displayed. Click **Next**
9. On the Summary screen, click **Create**.
10. On the Completion Summary screen, click **Close**.

Configuring Oracle Internet Directory With a WebLogic Domain

In this configuration, OID and a WebLogic Server domain is configured on the first host and the second host. The OID instance on the second host joins the domain created on the first host.

- Oracle Internet Directory Component Names Assigned by Oracle Identity Management Installer
  When you configure OID using the Config Wizard, the default instance that the installer assigns to the OID instance is `oid1`. You cannot change this name.
- Configuring Oracle Internet Directory on OIDHOST1
  Ensure that the schema database is running and that RCU has been used to seed the ODS database schema, then follow these steps to configure the OID instance on OIDHOST1:
- Configuring Oracle Internet Directory on OIDHOST2
  Ensure that the OID repository is running and then follow these steps to configure the OID instance on OIDHOST2:

Oracle Internet Directory Component Names Assigned by Oracle Identity Management Installer

When you configure OID using the Config Wizard, the default instance that the installer assigns to the OID instance is `oid1`. You cannot change this name.

The instance-specific configuration entry for this OID instance is `cn=oid1, cn=osdldapd, cn=subconfigsubentry`.

If you perform a second OID installation on another computer and that OID instance uses the same database as the first instance, the installer detects the previously installed OID instance on the other computer using the same Oracle database, so it gives the second OID instance a component name of `oid2`.

The instance-specific configuration entry for the second OID instance is `cn=oid2, cn=osdldapd, cn=subconfigsubentry`. Any change of properties in the entry `cn=oid2, cn=osdldapd, cn=subconfigsubentry` will not affect the first instance `(oid1)`.

If a third OID installation is performed on another computer and that instance uses the same database as the first two instances, the installer gives the third OID instance a component name of oid3, and so on for additional instances on different hosts that use the same database.

Note that the shared configuration for all OID instances is `cn=dsaconfig, cn=configsets,cn=oracle internet directory`. Any change in this entry will affect all the instances of OID.

Configuring Oracle Internet Directory on OIDHOST1

Ensure that the schema database is running and that RCU has been used to seed the ODS database schema, then follow these steps to configure the OID instance on OIDHOST1:

1. Ensure that the system, patch, kernel and other requirements are met. These are listed in Preparing to Install in *Oracle Fusion Middleware Installation Guide for Oracle Identity Management*guide.
2. Ensure that Oracle Identity Management software is installed and upgraded on OIDHOST1 Installing Oracle Fusion Middleware Components describes.
3. Ensure that ports 3060 and 3131 are not in use by any service on the computer by issuing these commands for the operating system you are using. If a port is not in use, no output is returned from the command.

   On UNIX:

   ```
   netstat -an | grep LISTEN | grep ":3060"
   netstat -an | grep LISTEN | grep ":3131"
   ```

   On Windows:

   ```
   netstat -an | findstr "LISTEN" | findstr ":3060"
   netstat -an | findstr "LISTEN" | findstr ":3131"
   ```

4. If the port is in use (if the command returns output identifying the port), you must free the port.
   a. On Unix:

      Remove any entries for ports 3060 and 3131 in the `/etc/services` file and restart the services, or restart the computer. You can also check for any existing processes that is using these ports, using `netstat -anp` command.

   b. On Windows:

      Stop the component that is using these ports.
5. Start the Configuration Wizard from `ORACLE_HOME/oracle_common/common/bin/config.sh` directory:

   On UNIX, issue this command: `./config.sh`

   On Windows, double-click `config.exe`
6. On **Create Domain** screen, select **Create a New Domain** and provide the domain location. Click **Next**.
7. On **Templates** screen, select `Oracle Internet Directory ( Collocated ) - 12.2.1.3.0 [oid]` template. Retain all the selected dependent templates. Click **Next**.
8. On **Administrator Account** screen, provide weblogic user password and click **Next**.
9. On **Domain Mode and JDK** screen, select **Production** in **Domain Mode** field and select **JDK8 Based Install** . Click **Next**.
10. On **Database Configuration Type** screen, specify the database connection parameters. Change the schema owner field value from `DEV_STB` to relevant prefix — `<PREFIX_STB>` — as needed, click **Get RCU Configuration** and click **Next**.
11. On **Component Datasources** screen, click **Next**.
12. On **JDBC Test** screen, after successful connection test, click **Next**.

13. On **Advanced Configuration** screen, select **Administration Server**, **Node Manager** and **Topology**. Click **Next**.
14. On **Administration Server** screen, update **Listen Address** to a desired hostname and **Listen Port** as needed. Click **Next**.
15. On **Node Manager Type** screen, provide *Node Manager credentials* and Click **Next**.
16. On **Manager Server**, **Skip** the screen and click **Next**.
17. On **Cluster** screen, **Skip** and click **Next**.
18. On **Server Templates** screen, **Skip** and click **Next**.
19. On **Coherence Clusters** screen, **Skip** and click **Next**.
20. On **Machines** screen, Do Not change the name of the default machine name as *oidhost1*. Update the **Listen Address** to appropriate host name. The Node Manager Listen Port can be changed, if required. This port number should be the same value as the one in `nodemanager.properties` file. Add a new machine with name - *oidhost2* and update the Listen Address to appropriate host name that points to OIDHOST2. If required, change the Listen Port to a desired port value.
21. On **Assign Servers to Machines** screen, select **oidhost1** and assign **AdminServer** to **oidhost1**. Click **Next**.
22. On **Virtual Targets** screen, click **Next**.
23. On **Partitions** screen, click **Next**.
24. On **Configuration Summary**, click **Create**.
25. Start **Administration Server**.
26. Start **Node Manager**.
27. From `ORACLE_HOME/oracle_common/common/bin/wlst.sh` directory, Run `wlst.sh` and execute the following commands:

    ```
    connect('weblogic','<password>', 't3://<admin-host>:<admin-port>')
    oid_setup(orcladminPassword='<desired-password>', odsPassword='ODS-
    schema-password")
    oid_createInstance(instanceName='oid2',
    machine='oidhost2',port='oid-non-ssl-port',sslPort='oid-ssl-port',
    host='hostname-of-OIDHOST2')
    exit()
    ```

28. From `ORACLE_HOME/oracle_common/common/bin/pack.sh` directory, execute `pack.sh` command, as shown below:

    ```
    pack.sh -domain=<DOMAIN_HOME_LOCATION> -template=./base_domain.jar -
    template_name=base_domain -managed=true
    ```

Configuring Oracle Internet Directory on OIDHOST2

Ensure that the OID repository is running and then follow these steps to configure the OID instance on OIDHOST2:

1. Ensure that the system, patch, kernel and other requirements are met. These are listed in Preparing to Install in *Oracle Fusion Middleware Installation Guide for Oracle Identity Management*guide.
2. Ensure that Oracle Identity Management software has been installed and upgraded on OIDHOST2 as described in Installing Oracle Fusion Middleware Components
3. On OIDHOST1, ports 3060 and 3131 were used for OID. The same ports should be used for the OID instance on OIDHOST2. Therefore, ensure that ports 3060 and 3131 are not in use by any service on OIDHOST2 by issuing these commands for the operating system you are using. If a port is not in use, no output is returned from the command.

   On Unix:

   ```
   netstat -an | grep LISTEN | grep ":3060"

   netstat -an | grep LISTEN | grep ":3131"
   ```

   On Windows:

   ```
   netstat -an | findstr "LISTEN" | findstr ":3060"

   netstat -an | findstr "LISTEN" | findstr ":3131"
   ```

4. If the port is in use (if the command returns output identifying the port), you must free the port.

   On Unix:
   Remove any entries for ports 3060 and 3131 in the `/etc/services` file and restart the services, or restart the computer. You can also check for any existing processes that is using these ports, using `netstat -anp` command.

   On Windows:
   Stop the component that is using these ports.

5. From `ORACLE_HOME/oracle_common/common` directory, create the domain using `unpack.sh` command. Use the packed domain jar file created in OIDHOST1:

   ```
   unpack.sh -template=./base_domain.jar -domain=<ORACLE_HOME>/user_projects/
   domains/base_domain
   ```

6. From the`DOMAIN_HOME/bin` directory, start Node Manager.

   ```
   ./startNodeManager.sh
   ```

7. From `DOMAIN_HOME/bin` directory, Start **oid2** instance by executing `startComponent.sh` script. Execute the script from **OIDHOST1** machine, where AdminServer is setup and not from OIDHOST2.

   - `./startComponent.sh oid2`
   - `oid2` can also be started either from OIDHOST1 or OIDHOST2 using WLST command — `nmStart()`

     ```
     nmStart(erverName='oid2', serverType='OID')
     ```

Validating Oracle Internet Directory High Availability

Use the `ldapbind` command-line tool to ensure that you can connect to each OID instance and the LDAP Virtual Server. The `ldapbind` tool enables you to determine whether you can authenticate a client to a server.

> **Note:**
>
> See the Configuring Your Environment section of *Oracle Fusion Middleware Reference for Oracle Identity Management* for a list of the environment variables you must set before using the`ldapbind` command.

For non-SSL:

```
ldapbind -h oidhost1.example.com -p 3060 -D "cn=orcladmin" -q
ldapbind -h oidhost2.example.com -p 3060 -D "cn=orcladmin" -q
ldapbind -h oid.example.com -p 3060 -D "cn=orcladmin" -q
```

> **Note:**
>
> The -q option prompts the user for a password. LDAP tools are modified to disable the options -w *password* and -P *password* when the environment variable LDAP_PASSWORD_PROMPTONLY is set to TRUE or 1. Use this feature whenever possible.

For SSL:

```
ldapbind -h oidhost1.example.com -p 3131 -D "cn=orcladmin" -q -U 1
ldapbind -h oidhost2.example.com -p 3131 -D "cn=orcladmin" -q -U 1
ldapbind -h oid.example.com -p 3131 -D "cn=orcladmin" -q -U 1
```

where -U is an optional argument used to specify the SSL authentication mode. These are the valid values for the SSL authentication mode:

- 1 = No authentication required
- 2 = One way authentication required. With this option, you must also supply a wallet location (-W "*file:/home/my_dir/my_wallet*") and wallet password (-P *wallet_password*).
- 3 = Two way authentication required. With this option, you must also supply a wallet location (-W "*file:/home/my_dir/my_wallet*") and wallet password (-P *wallet_password*).

For more information about the `ldapbind` command, see the ldapbind section in *Oracle Fusion Middleware Reference for Oracle Identity Management*.

For information about setting up SSL for OID, see Configuring Secure Sockets Layer (SSL) in the *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* manual.

WebLogic Server Administration Console:

```
http://oidhost1.example.com:7001/console
```

Oracle Enterprise Manager Fusion Middleware Console:

```
http://oidhost1.example.com:7001/em
```

Oracle Internet Directory Failover and Expected Behavior

This section describes how to perform a failover of Oracle Internet Directory and Oracle RAC.

This section includes the following topics:

- Performing Oracle Internet Directory Failover
  This procedure describes the steps to be followed to perform Oracle Internet Directory failover.
- Performing an Oracle RAC Failover
  The `orclfailoverenabled` attribute is a configuration entry `("cn=configset,cn=oidmon,cn=subconfigsubentry")` that configures failover for Oracle Internet Directory processes. This attribute specifies the failover time in minutes before the OID Monitor will start failed processes on a surviving node. The default failover time is 5 minutes. A value of zero (0) specifies that Oracle Internet Directory processes will not fail over to another node.

Performing Oracle Internet Directory Failover

This procedure describes the steps to be followed to perform Oracle Internet Directory failover.

The following example describes how to perform a failover to OIDHOST2 and check the status of OID service:

1. On OIDHOST1, use the following WLST command to stop the OID instance:

   ```
   shutdown(name='instance-name')
   ```

2. On OIDHOST2, check the status of OID using the load balancing router.

> **Note:**
>
> See the "Configuring Your Environment" section of Oracle Fusion Middleware Reference for Oracle Identity Management for a list of environment variables you must set before using the ldapbind command.

```
ldapbind -h oid.example.com -p 3060 -D "cn=orcladmin" -q
```

> **Note:**
>
> The -q option above prompts you for a password. LDAP tools are modified to disable the options -w password and -P password when the environment variable LDAP_PASSWORD_PROMPTONLY is set to TRUE or 1. Use this feature whenever possible.

Related Topics

- Managing Oracle Internet Directory Components by Using WLST Commands

Performing an Oracle RAC Failover

The `orclfailoverenabled` attribute is a configuration entry `("cn=configset,cn=oidmon,cn=subconfigsubentry")` that configures failover for Oracle Internet Directory processes. This attribute specifies the failover time in minutes before the OID Monitor will start failed processes on a surviving node. The default failover time is 5 minutes. A value of zero (0) specifies that Oracle Internet Directory processes will not fail over to another node.

To perform an Oracle RAC failover, perform the following steps:

1. Use the srvctl command to stop a database instance:

   ```
   srvctl stop instance -d db_unique_name -i inst_name_list
   ```

2. Use the srvctl command to check the status of the database:

   ```
   srvctl status database -d db_unique_name -v
   ```

3. Check the status of Oracle Internet Directory:

> **Note:**
>
> See **"Configuring Your Environment"** in **Oracle Fusion Middleware Reference for Oracle Identity Management** for a list of environment variables you must set before using the ldapbind command.

```
ldapbind -h oid_host1 -p 3060 -D "cn=orcladmin" -q
ldapbind -h oid_host2 -p 3060 -D "cn=orcladmin" -q
ldapbind -h oid.example.com -p 3060 -D "cn=orcladmin" -q
```

> **Note:**
>
> The -q option above prompts the user for a password. LDAP tools are modified to disable the options -w password and -P password when the environment variable LDAP_PASSWORD_PROMPTONLY is set to TRUE or 1. Use this feature whenever possible.

To know more about RAC failover, See Oracle Internet Directory Replication-Server Control and Failover

Troubleshooting Oracle Internet Directory High Availability

This section provides information that can help you troubleshoot OID high availability issues:

- Log files for OID are in directory:
  `DOMAIN_HOME/servers/OID/logs/InstanceName`
- The order in which log files should be examined when troubleshooting is:
  1. `oidmon-xxx.log`

2. `oiddispd01-xxxx.log`
3. `oidldapd01s-xxxx.log`

- This section shows some of the error messages that may be related to high availability, and their meaning:

  Error: ORA-3112, ORA-3113 errors in the log file

  Cause: one of the database node is down, OID connects again to surviving node.

  Action: See why database node went down or Oracle process got killed

  Error: Failing Over...Please stand by in the log file

  Cause: OID server received a notification from the Oracle process that one of the database node is down. OID will connect to the surviving node.

  – If the failover is successful you would see this message:

    Failover ended...resuming services.

  – If the failover was not successful, you would see these errors:

  – Tried 10 times, now quitting from failover function...

  – Bad Failover Event:

  – Forcing Failover abort as setting of DB parameters for the session failed

- If high availability event notification is enabled, you would see a message similar to the following:

```
HA Callback Event
Thread Id: 8
Event type: 0
HA Source: OCI_HA_INSTANCE
Host name: dbhost1
Database name: orcl
Instance name: orcl1
Timestamp: 14-MAY-09 03.25.24 PM -07:00
Service name: orcl.example.com
HA status: DOWN - TAF Capable
```

- If TAF is disabled, HA status will be shown as

  `DOWN.`

Action: See why database node went down.

Error: Time Difference of at least 250 sec found between node1 and node2.

Cause: There is time difference between the two nodes

Action: Synchronize the system time.

Error: Node=% did not respond for configured %d times, Failing over...

Cause: One of the OID nodes (oidmon) is not responding.

Action: See if the node is alive or OIDMON process is running.

Related Topics

- Troubleshooting Oracle Internet Directory

Additional Oracle Internet Directory High Availability Issues

This section describes issues for Oracle Internet Directory in a high availability environment.

This section describes issues for Oracle Internet Directory in a high availability environment.

See Changing the Password of the ODS Schema Used by Oracle Internet Directory

- Changing the Password of the ODS Schema Used by Oracle Internet Directory
  You can change the OID database schema password (that is, the password of the ODS user in the database) using the Oracle Internet Directory Database Password Utility (oidpasswd) from OIDHOST1 (Where AdminServer is installed). However, since the ODS schema password is stored in a password wallet under the DOMAIN_HOME on each host. This is propagated from OIDHOST1 to all other hosts automatically by Weblogic Domain Framework.

Changing the Password of the ODS Schema Used by Oracle Internet Directory

You can change the OID database schema password (that is, the password of the ODS user in the database) using the Oracle Internet Directory Database Password Utility (oidpasswd) from OIDHOST1 (Where AdminServer is installed). However, since the ODS schema password is stored in a password wallet under the DOMAIN_HOME on each host. This is propagated from OIDHOST1 to all other hosts automatically by Weblogic Domain Framework.

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

```
oidpasswd connect=database-connection-string change_oiddb_pwd=true
```

Oracle Directory Integration Platform High Availability

This section describes how to design and deploy a high availability environment for Oracle Directory Integration Platform (ODIP).

- Understanding Oracle Directory Integration Platform Component Architecture
  Oracle Directory Integration Platform is a J2EE application that enables you to integrate your applications and directories, including third-party LDAP directories, with an Oracle back-end directory: Oracle Internet Directory, Oracle Unified Directory, and Oracle Directory Server Enterprise Edition.
- Understanding Oracle Directory Integration Platform High Availability Concepts
  This section describes the Oracle Directory Integration Platform high availability concepts.
- Configuring Oracle Directory Integration Platform for High Availability
  You can use Oracle Internet Directory or Oracle Unified Directory as the as the back-end directory to configure Oracle Directory Integration Platform high availability.
- About Retrieving Changes from Connected Directories

- Understanding Oracle Directory Integration Platform Failover and Expected Behavior
  In a high availability environment, you deploy the Oracle Directory Integration Platform application on a WebLogic Server cluster that comprises at least two WebLogic instances.
- Troubleshooting Oracle Directory Integration Platform High Availability
  This section describes how to manage issues involving Oracle Directory Integration Platform high availability.

Understanding Oracle Directory Integration Platform Component Architecture

Oracle Directory Integration Platform is a J2EE application that enables you to integrate your applications and directories, including third-party LDAP directories, with an Oracle back-end directory: Oracle Internet Directory, Oracle Unified Directory, and Oracle Directory Server Enterprise Edition.

> **Note:**
>
> Oracle Directory Integration Platform does not support Oracle Directory Server Enterprise Edition in high availability mode in this release.

See Introduction to Oracle Directory Integration Platform in *Oracle Fusion Middleware Administering Oracle Directory* for more on Oracle Directory Integration Platform architecture.

Understanding Oracle Directory Integration Platform High Availability Concepts

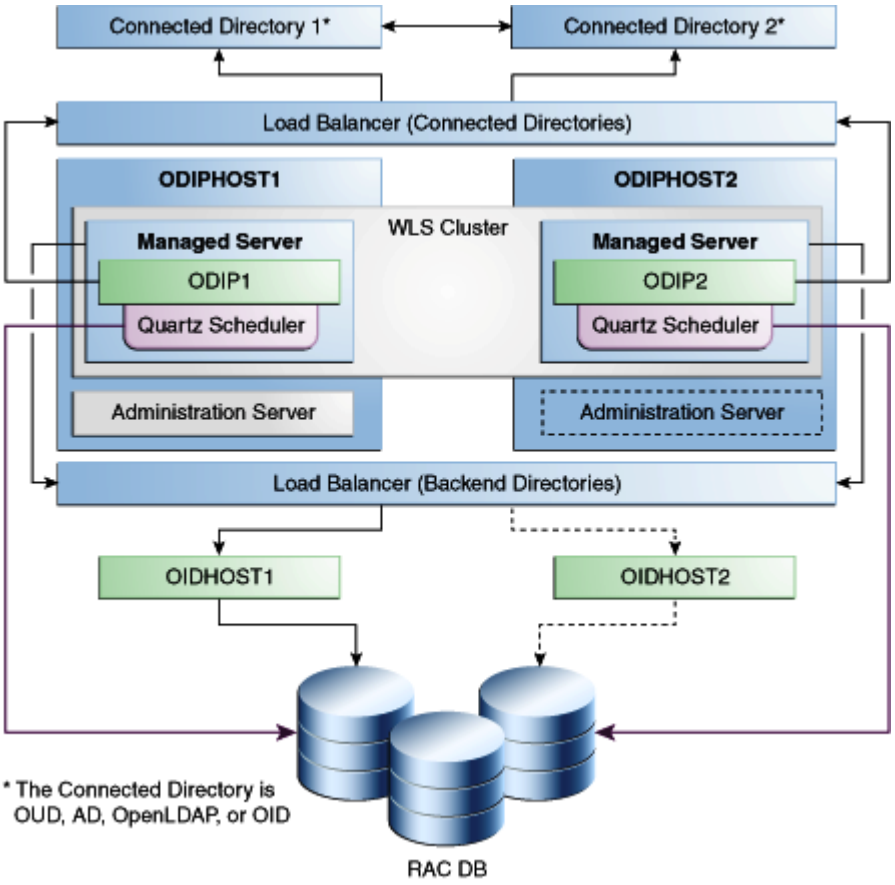
This section describes the Oracle Directory Integration Platform high availability concepts.

- About Oracle Directory Integration Platform High Availability Architecture (OID Back-End)
  Learn about the Oracle Directory Integration Platform high availability architecture with Oracle Internet Directory as the back-end directory.
- About Oracle Directory Integration Platform High Availability Architecture (OUD Back-End)
  This section describes the Oracle Directory Integration Platform high availability architecture with Oracle Unified Directory (OUD) as the back-end directory.
- Protection from Failures and Expected Behavior
  This section describes protection from different types of failure in an Oracle Directory Integration Platform active-active cluster

About Oracle Directory Integration Platform High Availability Architecture (OID Back-End)

Learn about the Oracle Directory Integration Platform high availability architecture with Oracle Internet Directory as the back-end directory.

Figure 12-2 Oracle Directory Integration Platform with Oracle Internet Directory (Back-End Directory) in a High Availability Architecture



In Figure 12-2 , Connected Directory 1 and Connected Directory 2 replicate information with each other. A load balancing router routes requests to the Connected Directories.

The Application Tier includes the `ODIPHOST1` and `ODIPHOST2` computers.

`ODIP1` and `ODIP2` go through the load balancer when they must communicate with the Connected Directories.

On `ODIPHOST1`, the following installations are performed:

- An Oracle Directory Integration Platform instance is installed (`ODIP1`) on the Managed Server.
- A Quartz Scheduler is installed on `ODIP1` by default. It connects to the Oracle RAC database using a WebLogic multi data source. The Quartz Scheduler invokes EJBs that do the actual work; if the EJB fails, the Quartz Scheduler marks the job as failed and reschedules it to run at later time by another EJB.
- An Administration Server is installed. Under normal operations, this is the active Administration Server.

On `ODIPHOST2`, the following installations are performed:

- An Oracle Directory Integration Platform instance is installed (`ODIP2`) on the Managed Server.
- A Quartz Scheduler is installed on `ODIP2` by default. Quartz Scheduler connects to the Oracle RAC database using a WebLogic multi data source.
- An Administration Server is installed. Under normal operations, this is the passive Administration Server instance. You make this Administration Server active if the Administration Server on `ODIPHOST1` becomes unavailable.

The Oracle Directory Integration Platform instances on the `ODIPHOST1` and `ODIPHOST2` Managed Servers are configured as a cluster.

A load balancer is set up for the back-end directories `OIDHOST1` and `OIDHOST2`. The load balancer routes requests to either OIDHOST1 or OIDHOST2.

> **Note:**
>
> When you use a RAC database, multi data source is used with Oracle Directory Integration Platform to protect the instances from RAC failure.

- About Starting and Stopping the Cluster
- Cluster-Wide Configuration Changes (OID)

About Starting and Stopping the Cluster

By default, the WebLogic Server starts, stops, and monitors the applications and Oracle Directory Integration Platform leverages the high availability features of the underlying clusters. If there is a hardware or other failure, session state is available to other cluster nodes that can resume the work of the failed node.

Node Manager monitors the WebLogic servers. If failure occurs, Node Manager restarts the WebLogic Server.

See General Node Manager Configuration in *Oracle Fusion Middleware Administering Node Manager for Oracle WebLogic Server*.

Cluster-Wide Configuration Changes (OID)

When you deploy Oracle Internet Directory in a high availability configuration, all Oracle Internet Directory instances in the cluster share the same database. Any changes made to Oracle Directory Integration Platform on one Oracle Internet Directory node automatically propagate to all the Oracle Internet Directory instances in the cluster.

Directory Synchronization Profiles

Changes that you make to directory integration profiles on one Oracle Internet Directory node do not replicate automatically to other Oracle Internet Directory nodes in a default multimaster Oracle Internet Directory replication environment. You must copy changes from the primary node to the secondary nodes manually and do so on a periodic basis. By doing this, a directory synchronization profile can run on a secondary node if a problem occurs on the primary node.

Oracle Directory Integration Platform uses the parameter `orcllastappliedchangenumber`. The value assigned to the `lastchangenumber` attribute in a directory synchronization profile

depends on the directory server on which Oracle Directory Integration Platform is running. In an active-active Oracle Directory Integration Platform configuration, you must manually update the `lastchangenumber` attribute in all instances.

To synchronize directory provisioning profiles between the primary Oracle Internet Directory node and secondary nodes:

1. On the primary node, use the `ldifwrite` command to create an LDIF dump of the entries from this container:

   ```
   cn=subscriber profiles,cn=changelog subscriber,cn=oracle internet directory
   ```

2. Copy the LDIF dump to the secondary node.
3. Use the `ldapadd` command to add the profiles on the secondary node.

After you copy an export profile to a target node, you must update the `lastchangenumber` attribute with the target node value. To update the value:

1. Disable the synchronization profile.
2. Get the value of the `lastchangenumber` attribute on the target node using the `ldapsearch` command.
3. Use `ldapsearch` to get the LDIF dump of the profile entry.
4. Use `ldapadd` to add the profile to the other Managed Server instance.
5. Go to the Oracle Directory Integration Platform Admin console and select the profile. Select **Edit**. Select the **Advanced** tab then select **Edit and Persist**. Enter the value of the `lastchangenumber` attribute. Save the profile.
6. Enable the synchronization profile.

Directory Provisioning Profiles

In a default multimaster Oracle Internet Directory replication environment, Oracle Directory Integration Platform is installed in the same location as the primary Oracle Internet Directory. The information and steps in this topic applies only when multimaster replication is set up.

If the primary node fails, event propagation stops for all profiles located on the node. Although the events are queued and not lost while the primary node is stopped, the events do not propagate to any applications that expect them. To ensure that events continue to propagate even when the primary node is down for the Version 1.0 and 2.0 profiles, the directory provisioning profiles must be copied to other secondary nodes.

However, copy directory provisioning profiles from the primary node to any secondary nodes immediately after an application is installed and before any user changes are made in Oracle Internet Directory.

To synchronize directory provisioning profiles between a primary node and any secondary nodes:

1. On the primary node, use the ldifwrite command to create an LDIF dump of the entries from this container:

   ```
   cn=provisioning profiles,cn=changelog subscriber,cn=oracle internet directory
   ```

2. Copy the LDIF dump to the secondary node.
3. Use the `ldapadd` command to add the profiles on the secondary node.

About Oracle Directory Integration Platform High Availability Architecture (OUD Back-End)

This section describes the Oracle Directory Integration Platform high availability architecture with Oracle Unified Directory (OUD) as the back-end directory.

Figure 12-3 Oracle Directory Integration Platform with Oracle Unified Directory (Back-End Directory) in a High Availability Architecture

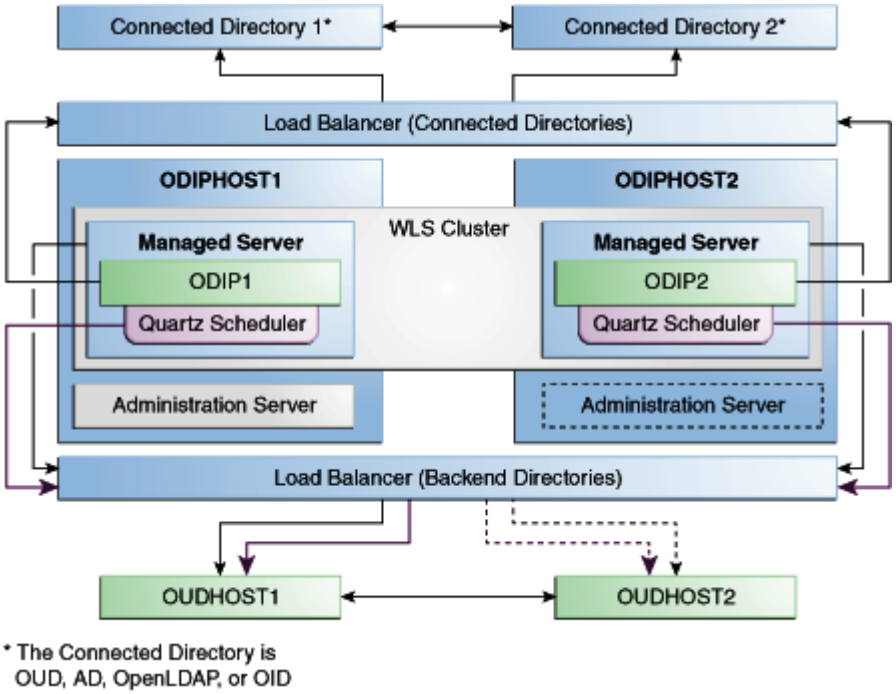


Figure 12-3, Connected Directory 1 and Connected Directory 2 replicate information with each other. A load balancing router routes requests to the Connected Directories.

The Application Tier includes the `ODIPHOST1` and `ODIPHOST2` computers.

On `ODIPHOST1`, the following installations are performed:

- An Oracle Directory Integration Platform instance is installed (`ODIP1`) on the Managed Server. `ODIP1` goes through the load balancer for connected directories when it must connect to them.
- The Quartz Scheduler is installed. It goes through the load balancer for the back-end directories.
- An Administration Server is installed. Under normal operations, this is the active Administration Server.

On `ODIPHOST2`, the following installations are performed:

- An ODIP instance is installed (`ODIP2`) on the Managed Server. `ODIP2` goes through the load balancer for connected directories when it must connect to them.

- The Quartz Scheduler is installed. It goes through the load balancer for backend directories.
- An Administration Server is installed. Under normal operations, this is the passive Administration Server instance. You make this Administration Server active if the Administration Server on `ODIPHOST1` becomes unavailable.

The Oracle Directory Integration Platform instances on the `ODIPHOST1` and `ODIPHOST2` Managed Servers are configured as a cluster.

A load balancer is set up for the back-end directories `OUDHOST1` and `OUDHOST2`. The load balancer routes requests to either `OUDHOST1` or `OUDHOST2`.

- Cluster-Wide Configuration Changes (OUD)

Cluster-Wide Configuration Changes (OUD)

Oracle Unified Directory supports cluster-wide configuration changes. All Oracle Unified Directory instances that are part of the same replication topology share the same content. Any changes made to Oracle Directory Integration Platform on one Oracle Unified Directory node automatically propagate to all Oracle Unified Directory instances in the replication topology.

Protection from Failures and Expected Behavior

This section describes protection from different types of failure in an Oracle Directory Integration Platform active-active cluster

- About Process Failure
- About Updating the Oracle Directory Integration Platform Server Configuration
- About External Dependency Failure

About Process Failure

In a high availability environment, you deploy the Oracle Directory Integration Platform application to a cluster that comprises at least two Oracle WebLogic instances.

By default, the Oracle Directory Integration Platform application leverages high availability features of the underlying WebLogic clusters. When you deploy Oracle Directory Integration Platform, the Quartz scheduler starts with a clustering option. Depending on the load on the node, the scheduler then runs the job on any available nodes in the cluster. If hardware or other failures occur on one or more nodes, the Quartz scheduler runs the jobs on available nodes.

Also, Node Manager monitors WebLogic servers. In case of failure, Node Manager restarts the WebLogic server.

Within the Oracle Directory Integration Platform application, the Quartz Scheduler invokes the Provisioning or Synchronization EJBs that do the actual work. As soon as the Quartz scheduler invokes an EJB, it tags that EJB as running the job. If the EJB fails, the Quartz scheduler marks the job as failed and reschedules it to run later by another EJB.

About Updating the Oracle Directory Integration Platform Server Configuration

If the back-end server is not accessed or cannot be accessed through a load balancer, Oracle Directory Integration Platform failover is not transparent.

This scenario requires manual intervention because the information to connect to the back-end directory is local to each Oracle Directory Integration Platform instance.

You must run the `manageDIPServerConfig` utility to update the Oracle back-end directory (Oracle Internet Directory and Oracle Unified Directory) `host` and `port` parameters for all of the Oracle Directory Integration Platform instances.

See manageDIPServerConfig Utility in *Oracle Fusion Middleware Administering Oracle Directory Integration Platform*.

About External Dependency Failure

Oracle Directory Integration Platform requires the back-end repository, Oracle Internet Directory, Oracle Unified Directory, Credential Store Framework, and the WebLogic Managed Server to be available during startup.

It fails to start if any one of these elements are unavailable.

Configuring Oracle Directory Integration Platform for High Availability

You can use Oracle Internet Directory or Oracle Unified Directory as the as the back-end directory to configure Oracle Directory Integration Platform high availability.

- Configuring High Availability for an Oracle Internet Directory Back-End Server
  Use the steps in the following order to configure Oracle Internet Directory (back-end directory) for Oracle Directory Integration Platform high availablity.
- Configuring High Availability for an Oracle Unified Directory Back-End Server
  Use the steps in the following order to configure Oracle Unified Directory (back-end directory) for Oracle Directory Integration Platform high availability.

Configuring High Availability for an Oracle Internet Directory Back-End Server

Use the steps in the following order to configure Oracle Internet Directory (back-end directory) for Oracle Directory Integration Platform high availablity.

- Before You Configure Oracle Directory Integration High Availability (OID)
- Configuring Oracle Directory Integration Platform on ODIPHOST1 (OID)
- Configuring Oracle Directory Integration Platform for Oracle Internet Directory (OIDHOST1)
- Configuring Oracle Directory Integration Platform on ODIPHOST2 (OID)
- Before You Configure Oracle Directory Integration High Availability (OID)
- Configuring Oracle Directory Integration Platform on ODIPHOST1 (OID)
- Configuring Oracle Directory Integration Platform for Oracle Internet Directory (OIDHOST1)
  You must configure Oracle Directory Integration Platform for Oracle Internet Directory on `OIDHOST1` instance.

- Configuring Oracle Directory Integration Platform on ODIPHOST2 (OID)

Before You Configure Oracle Directory Integration High Availability (OID)

Complete the following before you configure Oracle Directory Integration Platform high availability with Oracle Internet Directory as the back-end directory:

- Ensure that Oracle Internet Directory is configured for high availability, as described in Oracle Internet Directory High Availability Configuration Steps.
- Oracle WebLogic Server and Oracle Directory Integration Platform is installed across all nodes (`ODIPHOST1` and `ODIPHOST2`).

Configuring Oracle Directory Integration Platform on ODIPHOST1 (OID)

To configure Oracle Directory Integration Platform on `ODIPHOST1`:

1. Start the Configuration Wizard by running the `ORACLE_HOME/oracle_common/common/bin/config.sh` script (on UNIX) or `ORACLE_HOME\oracle_common\common\bin\config.cmd` (on Windows).

   The **Configuration Type** screen is displayed.

2. Select **Update an existing domain**, and click **Next**.

   The **Templates** screen is displayed.

3. On the **Templates** screen, select **Update Domain Using Product Templates** and then select **Oracle Directory Integration Platform - 12.2.1.3.0[dip]** domain configuration option.

   > **Note:**
   >
   > When you select the **Oracle Directory Integration Platform - 12.2.1.3.0 [dip]** option, **Oracle Enterprise Manager 12.2.1.3.0 [em]** is automatically selected.

   Click **Next**.

   The **JDBC Data Sources** screen is displayed.

4. Make changes if required and then click **Next**

   The **JDBC Data Sources Test** screen is displayed.

5. Select the data sources to test, and click **Test Selected Connections**.

   Click **Next**.

   The **Database Configuration Type** screen is displayed.

6. Make changes if required and then click **Get RCU Configuration** to retrieve the schema information. After successfully retrieving the schema information, click **Next** to continue.

   The **JDBC Component Schema** screen is displayed.

7. Verify that the values populated are correct for all schemas, and Click **Next**.

> **Note:**
>
> To convert one or more of the schemas to Oracle RAC multi-data source schemas, select the check boxes next to the name of those schemas, and select the **Convert to RAC multi data source** option. Click Next when done. When you click Next, the **Oracle RAC Multi Data Source Component Schema** screen appears.
>
> See Oracle RAC Multi Data Source Component Schema in *Oracle Fusion Middleware Creating WebLogic Domains Using the Configuration Wizard*.

The **JDBC Component Schema Test** screen is displayed.

8. You can select the component schema to test, and click **Test Selected Connections**. Wait for one or more connection tests to complete. If you do not want to test connections, deselect all data sources.

> **Note:**
>
> In order to test connections, the database to which you are trying to connect must be running.

Click **Next**.

The **Advanced Configuration** screen is displayed.

9. Select **Managed Servers, Clusters, and Coherence** option. Click **Next**.

The **Managed Servers** screen is displayed.

10. Click **Add**, and create one Managed Servers each for `ODIPHOST1` and `ODIPHOST2`.

Table 12-9 Managed Server on ODIPHOST1

Name	Listen Address	Listen Port
`wls_ods1`	`odipHost1.example.com`	7005

Table 12-10 Managed Server on ODIPHOST2

Name	Listen Address	Listen Port
`wls_ods2`	`odipHost2.example.com`	7005

Click **Next**.

The **Clusters** screen is displayed.

11. Click **Add** and enter `odip_cluster` in the **Cluster Name** field to configure cluster for the Managed Servers on `ODIPHOST1` and `ODIPHOST2`.

Click **Next**.

The **Server Templates** screen is displayed.

12. Click **Next** and **Dynamic Servers** screen is displayed.

    Click **Next**.

    The **Assign Servers to Clusters** screen is displayed.

13. Use the Assign Servers to Clusters screen to assign the `wls_ods1` and `wls_ods2` Managed Servers to the `odip_cluster` cluster. Only Managed Servers appear in the Server list box. The Administration Server is not listed because it cannot be assigned to a cluster.

    Select the name of the Managed Server in the **Servers** list box and click the right arrow. The name of the Managed Server is removed from the **Servers** list box and added below the name of the target cluster in the **Clusters** list box.

    The name of the Managed Server is removed from the Servers list box and added below the name of the target cluster in the Clusters list box.

    Click **Next** and continue clicking **Next** till the **Machines** screen is displayed.

14. Click the **Machine** (for Windows) or **Unix Machine** tab (for UNIX) and then click **Add** to add the following machines:

    Table 12-11 Machines

Name	Node Manager Listen Address	Node Manager Listen Port
`odip_1`	`odipHost1.example.com`	`5556`
`odip_2`	`odipHost2.example.com`	`5556`

    Click **Next**.

    The **Assign Servers to Machines** screen is displayed.

15. Use the **Assign Servers to Machines** to assign the WebLogic Server instances to each of the machines.

    a. In the **Machine** list box, select the `odip_1` machine.

    b. Select the `wls_ods1` instance in the **Server** list box and click the right arrow.

       The name of the `wls_ods1` instance is removed from the **Server** list box and added, below the name of the target machine, in the **Machine** list box.

    c. Repeat above steps to assign `odip_2` machine to the `wls_ods2` Managed Server.

    Select the name of the Managed Server in the **Servers** list box and click the right arrow. The name of the Managed Server is removed from the **Servers** list box and added below the name of the target cluster in the **Clusters** list box.

    The name of the Managed Server is removed from the Servers list box and added below the name of the target cluster in the Clusters list box.

    Click **Next** and continue clicking **Next** till the **Configuration Summary** screen is displayed.

16. Review each item on the **Configuration Summary** screen and verify that the information is correct.

To make any changes, go back to a screen by clicking the Back button or selecting the screen in the navigation pane. Domain creation does not start until you click **Create**.

A new WebLogic domain (for example: *base_domain*) is created to support Oracle Directory Integration Platform and Fusion Middleware Control in the `<ORACLE_HOME>\user_projects\domains` directory (on Windows). On UNIX, the domain is created in the `<ORACLE_HOME>/user_projects/domains` directory.

Configuring Oracle Directory Integration Platform for Oracle Internet Directory (OIDHOST1)

You must configure Oracle Directory Integration Platform for Oracle Internet Directory on `OIDHOST1` instance.

Complete the following steps:

1. Run the `dipConfigurator` command to configure Oracle Directory Integration Platform (`ODIPHOST1`) for `OIDHOST1`. For more information, see Configuring Oracle Directory Integration Platform for Oracle Internet Directory in *Oracle Fusion Middleware Administering Oracle Directory Integration Platform*.

   > **Note:**
   >
   > - If you are using a RAC database, then Oracle recommends that you specify the URL for the RAC database in the `dbconfigfile` file for `dipConfigurator` properties.
   > - If the cipher suites configured for Oracle Internet Directory are not available or recognized in Oracle Directory Integration Platform then you must add those suites into Oracle Directory Integration Platform using the Oracle Fusion Middleware System MBean Browser. See Adding Cipher Suites Configured for Oracle Internet Directory into Oracle Directory Integration Platformin *Oracle Fusion Middleware Administering Oracle Directory Integration Platform*.

2. Run the `manageDIPServerConfig` command to tune the cluster:

   ```
   ./manageDIPServerConfig set -host ODIPHOST1.example.com -port 7005 -wlsuser
   weblogic -attribute ClusterCheckInInterval -value 30000

   ./manageDIPServerConfig set -host ODIPHOST1 -port 7005 -wlsuser weblogic -
   attribute RefreshInterval -value 120
   ```

3. Run the `manageDIPServerConfig` command for reconfiguring Oracle Directory Integration Platform to use the TCP load balancer.

   `LB_HOST` is the load balancer IP address you must configure to redirect to one of the back-end instances.

   ```
   ./manageDIPServerConfig set -host ODIPHOST1 -port 7005 -wlsuser weblogic -
   attribute BackendHostPort -value LB_HOST:LB_PORT
   ```

Configuring Oracle Directory Integration Platform on ODIPHOST2 (OID)

You must configure the Oracle Directory Integration Platform on `ODIPHOST2` for the Oracle Internet Directory back-end directory:

1. Run the following pack command on `ODIPHOST1` to create a template pack:

```
cd MW_HOME/oracle_common/common/bin
./pack.sh -managed=true -domain=MW_HOME/user_projects/domains/domainName -
template=dipdomain.jar -managed=true -template_name="dipdomain"
```

2. Copy the template file created in the previous step from `ODIPHOST1` to `ODIPHOST2`. For example, on a UNIX platform:

```
scp dipdomain.jar user@ODIPHOST2:MW_HOME/oracle_common/common/bin
```

3. Perform the following on `ODIPHOST2`:

   a. Run the `unpack` command to unpack the propagated template:

```
cd MW_HOME/oracle_common/common/bin
./unpack.sh -domain=MW_HOME/user_projects/domains/domains/domainName -
template=dipdomain.jar -overwrite_domain=true
```

   b. Start and stop the `wls_ods2` Managed Server:

```
MW_HOME/user_projects/domains/domainName/bin/startManagedWebLogic.sh
wls_ods2 http://ODIPHOST1:ODIPHOST1ADMINPORT

MW_HOME/user_projects/domains/domainName/bin/stopManagedWebLogic.sh
wls_ods2 http://ODIPHOST1:ODIPHOST1ADMINPORT
```

   c. Overwrite the `dip-config.xml` file in `wls_ods2` with the `dip-config.xml` in `wls_ods1`:

```
cp MW_HOME/user_projects/domains/DOMAIN_NAME/config/fmwconfig/servers/
wls_ods1/applications/DIP_12.2.1.3.0/configuration/dip-config.xml
 MW_HOME/user_projects/domains/DOMAIN_NAME/config/fmwconfig/servers/
wls_ods2/applications/DIP_12.2.1.3.0/configuration/dip-config.xml
```

   d. Start the Node Manager, by running the `startNodeManager.cmd` (Windows) or `startNodeManager.sh` (UNIX) command.

```
MW_HOME/user_projects/domains/DOMAIN_NAME/bin/startNodeManager.sh
```

   e. Start the `wls_ods2` Managed Server:

```
MW_HOME/user_projects/domains/DOMAIN_NAME/bin/startManagedWebLogic.sh
wls_ods2 http://ODIPHOST1:ODIPHOST1ADMINPORT
```

Configuring High Availability for an Oracle Unified Directory Back-End Server

Use the steps in the following order to configure Oracle Unified Directory (back-end directory) for Oracle Directory Integration Platform high availability.

- Before You Configure Oracle Directory Integration High Availability (OUD)
- Configuring Oracle Directory Integration Platform on ODIPHOST1 (OUD)
- Configuring Oracle Directory Integration Platform for Oracle Unified Directory (OUDHOST1)
  You must configure Oracle Directory Integration Platform for Oracle Unified Directory on `OIDHOST1` instance.
- Configuring Oracle Directory Integration Platform on ODIPHOST2 (OUD)

Before You Configure Oracle Directory Integration High Availability (OUD)

Complete the following before you configure Oracle Directory Integration Platform high availability with Oracle Unified Directory as the back-end directory:

- Ensure that you install Oracle Unified Directory, see Installing the Oracle Unified Directory Software in *Oracle Fusion Middleware Installing Oracle Unified Directory*.

  When you set up an Oracle Unified Directory server instance using either the graphical user interface (GUI) or the command-line interface (CLI), ensure that you select the **Enable for DIP** option to enable the server instance for Oracle Directory Integration Platform.
- Ensure that Oracle Unified Directory is configured for high availability. See Understanding Oracle Unified Directory High Availability Deployments in *Oracle Fusion Middleware Administering Oracle Unified Directory*.
- Ensure that you have created the Oracle Unified Directory Suffixes for Oracle Directory Integration Platform. See Creating Oracle Unified Directory Suffixes in *Oracle Fusion Middleware Administering Oracle Directory Integration Platform*.
- Ensure that the change log is enabled. See Enabling External Change Login *Oracle Fusion Middleware Administering Oracle Directory Integration Platform*.
- Oracle WebLogic Server and Oracle Directory Integration Platform is installed across all nodes (`ODIPHOST1` and `ODIPHOST2`).

Configuring Oracle Directory Integration Platform on ODIPHOST1 (OUD)

To configure Oracle Directory Integration Platform on `ODIPHOST1` for Oracle Unified Directory as the back-end directory:

1. Start the Configuration Wizard by running the `<MW_HOME>/oracle_common/common/bin/config.sh` script (on UNIX) or `<MW_HOME>\oracle_common\common\bin\config.cmd` (on Windows).

   The **Configuration Type** screen is displayed.
2. On the **Configuration Type** screen, select **Create a new domain** and enter the full path for the domain or use the **Browse** button to navigate to the directory in which your domains are located. Click **Next**.

   The **Templates** screen is displayed.
3. On the **Templates** screen, make sure **Create Domain Using Product Templates** is selected, and then select **Oracle Directory Integration Platform - 12.2.1.3.0 [dip]**.

   > **Note:**
   >
   > When you select **Oracle Directory Integration Platform - 12.2.1.3.0 [dip]** option, the following components are automatically selected:
   >
   > - **Oracle Enterprise Manager 12.2.1.3.0 [em]**
   > - **Oracle JRF - 12.2.1.3.0 [oracle_common]**
   > - **Weblogic Coherence Cluster Extension 12.2.1.3 [wlserver]**

   Click **Next**.

   Click The **Application Location** screen is displayed.
4. Click **Browse** and specify the full path to the directory in which you want to store the applications that are associated with the domain.

Click **Next**.

The **Administrator Account** screen is displayed.

5. Specify the user name and password for the default WebLogic Administrator account for the domain.

   The password must be at least eight characters and must contain at least one number or special character. Confirm the password and click **Next**.

   Make a note of these details as you will need them to start or restart the WebLogic domain in the following procedure.

   The **Domain Mode and JDK** screen is displayed.

6. Specify the domain mode and Java Development Kit (JDK).

   a. Select **Production** in the Domain Mode field.

   > **Note:**
   >
   > If you select **Production** mode as the domain, the node manager has a random username and password assigned to it. Use the WebLogic Server Administration Console to reset the password.

   b. Accept **Oracle Hotspot** as a default JDK location.

   c. Click **Next**.

   The **Database Configuration Type** screen is displayed.

7. Select **RCU Data**. This option instructs the Configuration Wizard to connect to the database's Service Table (STB) schema to automatically retrieve schema information for schemas needed to configure the domain.

   > **Note:**
   >
   > Ensure that you have created the database schemas required for Oracle Internet Directory. See Creating the Database Schemas in *Oracle Fusion Middleware Installing and Configuring Oracle Internet Directory*.

   After selecting RCU Data:

   a. Enter the name of the server hosting the database in the **Host Name** field.

   b. Enter the database DBMS name, or service name if you selected a service type driver in the **DBMS/Service** field.

   c. Enter the port number on which the database listens.

   d. Enter the username and password for connecting to the database's Service Table schema.

   e. Click **Get RCU Configuration** to retrieve the schema information. After successfully retrieving the schema information, click **Next** to continue.

   The **JDBC Component Schema** screen is displayed.

8. Verify that the values populated are correct for all schemas, and Click **Next**.

> **Note:**
>
> To convert one or more of the schemas to Oracle RAC multi-data source schemas, select the check boxes next to the name of those schemas, and select the **Convert to RAC multi data source** option. Click Next when done. When you click Next, the **Oracle RAC Multi Data Source Component Schema** screen appears.
>
> See Oracle RAC Multi Data Source Component Schema in *Oracle Fusion Middleware Creating WebLogic Domains Using the Configuration Wizard*.

The **JDBC Component Schema Test** screen is displayed.

9. Click **Test Selected Connection** to test datasource connections that you just configured.

   A green check mark in the Status column indicates a successful test. If you encounter issues, see the error message in the Connection Result Log section of the screen, fix the problem, then test the connection again.
   The **Advanced Configuration** screen is displayed.

10. To complete domain configuration, select the following options:

    - **Administration Server**: Required to properly configure the Administration Server's listen address.
    - **Node Manager**: Required to configure Node Manager.
    - **Topology**: Required to configure the Managed Servers and cluster, and for configuring the machine and targeting Managed Servers to the machine.

    Click **Next**.

    The **Administration Server** screen is displayed.

11. Accept the default settings or change the Administration Server settings.

    Click **Next**.

    The **Node Manager** screen is displayed.

12. Use the Node Manager screen to select the Node Manager configurations that are applicable for the domain and click **Next**.

    The **Managed Servers** screen is displayed.

13. Click **Add**, and create one Managed Servers each for `ODIPHOST1` and `ODIPHOST2`.

Table 12-12 Managed Servers on ODIPHOST1

Name	Listen Address	Listen Port
`wls_ods1`	`odipHost1.example.com`	7005

Table 12-13 Managed Servers on ODIPHOST2

Name	Listen Address	Listen Port
`wls_ods2`	`odipHost2.example.com`	7005

Click **Next**.

The **Clusters** screen is displayed.

14. Click **Add** and enter `odip_cluster` in the **Cluster Name** field to configure cluster for the Managed Servers on `ODIPHOST1` and `ODIPHOST2`.

    Click **Next**.

    The **Server Templates** screen is displayed.

15. Click **Next** and **Dynamic Servers** screen is displayed.

    Click **Next**.

    The **Assign Servers to Clusters** screen is displayed.

16. Use the Assign Servers to Clusters screen to assign the `wls_ods1` and `wls_ods2` Managed Servers to the `odip_cluster` cluster. Only Managed Servers appear in the Server list box. The Administration Server is not listed because it cannot be assigned to a cluster.

    Select the name of the Managed Server in the **Servers** list box and click the right arrow. The name of the Managed Server is removed from the **Servers** list box and added below the name of the target cluster in the **Clusters** list box.

    The name of the Managed Server is removed from the Servers list box and added below the name of the target cluster in the Clusters list box.

    Click **Next** and continue clicking **Next** till the **Machines** screen is displayed.

17. Click the **Machine** or **Unix Machine** tab and then click **Add** to add the following machines:

Table 12-14 Machines

Name	Node Manager Listen Address	Node Manager Listen Port
`odip_1`	`odipHost1.example.com`	`5556`
`odip_2`	`odipHost2.example.com`	`5556`

    Click **Next**.

    The **Assign Servers to Machines** screen is displayed.

18. Use the **Assign Servers to Machines** to assign the WebLogic Server instances to each of the machines.

    a. In the **Machine** list box, select the `odip_1` machine.

    b. Select the `wls_ods1` instance in the **Server** list box and click the right arrow.

       The name of the `wls_ods1` instance is removed from the **Server** list box and added, below the name of the target machine, in the **Machine** list box.

    c. Repeat above steps to assign `odip_2` machine to the `wls_ods2` Managed Server.

    Select the name of the Managed Server in the **Servers** list box and click the right arrow. The name of the Managed Server is removed from the **Servers** list box and added below the name of the target cluster in the **Clusters** list box.

    The name of the Managed Server is removed from the Servers list box and added below the name of the target cluster in the Clusters list box.

Click **Next** and continue clicking **Next** till the **Configuration Summary** screen is displayed.

19. Review each item on the **Configuration Summary** screen and verify that the information is correct.

    To make any changes, go back to a screen by clicking the Back button or selecting the screen in the navigation pane. Domain creation does not start until you click **Create**.

    A new WebLogic domain (for example: *base_domain*) is created to support Oracle Directory Integration Platform and Fusion Middleware Control in the `<MW_HOME>\user_projects\domains` directory (on Windows). On UNIX, the domain is created in the `<MW_HOME>/user_projects/domains` directory.

Configuring Oracle Directory Integration Platform for Oracle Unified Directory (OUDHOST1)

You must configure Oracle Directory Integration Platform for Oracle Unified Directory on `OIDHOST1` instance.

Complete the following steps:

1. Run the `dipConfigurator` command to configure Oracle Directory Integration Platform (`ODIPHOST1`) for `OUDHOST1`. For more information, see Configuring Oracle Directory Integration Platform for Oracle Unified Directory in *Oracle Fusion Middleware Administering Oracle Directory Integration Platform*.
2. Run the `manageDIPServerConfig` command to tune the cluster:

   ```
   ./manageDIPServerConfig set -host ODIPHOST1.example.com -port 7005 -wlsuser
   weblogic -attribute ClusterCheckInInterval -value 30000

   ./manageDIPServerConfig set -host ODIPHOST1 -port 7005 -wlsuser weblogic -
   attribute RefreshInterval -value 120
   ```

3. Run the `manageDIPServerConfig` command for reconfiguring Oracle Directory Integration Platform to use the TCP load balancer.

   `LB_HOST` is the load balancer IP address you must configure to redirect to one of the back-end instances.

   ```
   ./manageDIPServerConfig set -host ODIPHOST1 -port 7005 -wlsuser weblogic -
   attribute BackendHostPort -value LB_HOST:LB_PORT
   ```

Configuring Oracle Directory Integration Platform on ODIPHOST2 (OUD)

You must configure the Oracle Directory Integration Platform on `ODIPHOST2` for the Oracle Unified Directory back-end directory:

1. Run the following pack command on `ODIPHOST1` to create a template pack:

   ```
   cd MW_HOME/oracle_common/common/bin
   ./pack.sh -managed=true -domain=MW_HOME/user_projects/domains/domainName -
   template=dipdomain.jar -managed=true -template_name="dipdomain"
   ```

2. Copy the template file created in the previous step from `ODIPHOST1` to `ODIPHOST2`. For example, on a UNIX platform:

   ```
   scp dipdomain.jar user@ODIPHOST2:MW_HOME/oracle_common/common/bin
   ```

3. Perform the following on `ODIPHOST2`:

   a. Run the `unpack` command to unpack the propagated template:

```
cd MW_HOME/oracle_common/common/bin
./unpack.sh -domain=MW_HOME/user_projects/domains/domains/domainName -
template=dipdomain.jar -overwrite_domain=true
```

b. Start and stop the `wls_ods2` Managed Server:

```
MW_HOME/user_projects/domains/domainName/bin/startManagedWebLogic.sh
wls_ods2 http://ODIPHOST1:ODIPHOST1ADMINPORT

MW_HOME/user_projects/domains/domainName/bin/stopManagedWebLogic.sh
wls_ods2 http://ODIPHOST1:ODIPHOST1ADMINPORT
```

c. Overwrite the `dip-config.xml` file in `wls_ods2` with the `dip-config.xml` in `wls_ods1`:

```
cp MW_HOME/user_projects/domains/DOMAIN_NAME/config/fmwconfig/servers/
wls_ods1/applications/DIP_12.2.1.3.0/configuration/dip-config.xml
 MW_HOME/user_projects/domains/DOMAIN_NAME/config/fmwconfig/servers/
wls_ods2/applications/DIP_12.2.1.3.0/configuration/dip-config.xml
```

d. Start the Node Manager, by running the `startNodeManager.cmd` (Windows) or `startNodeManager.sh` (UNIX) command.

```
MW_HOME/user_projects/domains/DOMAIN_NAME/bin/startNodeManager.sh
```

e. Start the `wls_ods2` Managed Server:

```
MW_HOME/user_projects/domains/DOMAIN_NAME/bin/startManagedWebLogic.sh
wls_ods2 http://ODIPHOST1:ODIPHOST1ADMINPORT
```

About Retrieving Changes from Connected Directories

Oracle Directory Integration Platform uses readers to retrieve changes from connected directories. However, there are some connectors that you cannot use for load-balanced directories. This section describes how Oracle Directory Integration Platform supports the use of several instances of a connected directory for import profiles.

Failing Over Oracle Directory Server Enterprise Edition Manually

Oracle Directory Integration Platform does not support transparent failover from one Oracle Directory Server Enterprise Edition (ODSEE) Managed Server (WLS_ODSEE1) to another ODSEE server (WLS_ODSEE2). Even if you replicate ODSEE Managed Server instances, the change numbers may not be identical on both ODSEE Managed Servers for the same update. If Oracle Directory Integration Platform fails over transparently from WLS_ODSEE1 to WLS_ODSEE2, ODIP may replay changes or miss changes each time it switches.

Oracle Unified Directory

When you use Oracle Unified Directory with Iplanet Reader and Iplanet Writer, Oracle Unified Directory does not support transparent failover from one Oracle Unified Directory instance to another because, as with ODSEE Server, change numbers may not be synchronized. However, you can configure your profile to use an Oracle Unified Directory connector that does support it.

To configure your profile, you must set the reader to `oracle.ldap.odip.gsi.OudCookieReader`. You must configure this attribute at creation time; you cannot configure it for existing profiles.

1. Go to the directory `ORACLE_HOME/ldap/odi/conf` and edit the file `iplanetimp.cfg.master`
2. Replace the line `Reader: oracle.ldap.odip.gsi.IPlanetReader` with the line `oracle.ldap.odip.gsi.OudCookieReader`

To failover transparently from one Oracle Unified Directory instance to another, the reader uses the External Change Log cookie that Oracle Unified Directory provides. The last applied change number contains a cookie but no longer contains a change number.

See Using the External Change Log in *Oracle Fusion Middleware Administering Oracle Unified Directory* for more information on Oracle Unified Directory external change log cookies.

Novell eDirectory

Because the Oracle Directory Integration Platform reader for Novell eDirectory is based on timestamps, clocks on all instances must be synchronized.

OpenLDAP

Because Oracle Directory Integration Platform reader for OpenLDAP is based on timestamps, clocks on all instances must be synchronized.

IBM Tivoli Directory Server

Oracle does not support IBM Tivoli by means of the load balancer.

Oracle Internet Directory

If you configure Oracle Internet Directory replication so that change numbers are identical on all Oracle Internet Directory instances that you target, the Oracle Internet Directory instances can failover transparently. If you do not set up this configuration, transparent failover is not supported.

Understanding Oracle Directory Integration Platform Failover and Expected Behavior

In a high availability environment, you deploy the Oracle Directory Integration Platform application on a WebLogic Server cluster that comprises at least two WebLogic instances.

By default, the Oracle Directory Integration Platform application leverages high availability features of the underlying WebLogic clusters. In case of hardware or other failures, session state is available to other cluster nodes that can resume the work of the failed node.

In addition, in a high availability environment, Node Manager is configured to monitor the WebLogic servers. In case of failure, Node Manager restarts the WebLogic Server.

If an instance of Oracle Internet Directory fails, the load balancer redirects to the surviving instance of Oracle Internet Directory and the Oracle RAC database. If Oracle Unified Directory fails, the load balancer redirects to the surviving instance of Oracle Unified Directory.

In case of a database instance failure, the surviving Oracle RAC node takes over any remaining processes. There may be innocuous errors in the Managed Servers logs during an Oracle RAC failover; see Troubleshooting Oracle Directory Integration Platform High Availability.

Troubleshooting Oracle Directory Integration Platform High Availability

This section describes how to manage issues involving Oracle Directory Integration Platform high availability.

Managed Server Log File Exception May Occur During an Oracle RAC Failover

During an Oracle RAC failover, exceptions similar to the ones below are seen in the Managed Server log files running the Oracle Directory Integration Platform application. These errors are thrown when the multi data sources configured on the WebLogic Server platform try to verify the health of the Oracle RAC database instances during failover. These are innocuous errors that you can ignore. The Oracle Directory Integration Platform application recovers and begins to operate normally after a lag of one or two minutes. During an Oracle RAC failover, there will be no Oracle Directory Integration Platform down time if one Oracle RAC instance is running at all times.

```
RuntimeException:
[2008-11-21T00:11:10.915-08:00] [wls_ods] [ERROR] []
[org.quartz.impl.jdbcjobstore.JobStoreTX] [tid: 25] [userId: <anonymous>]
[ecid: 0000Hqy69UiFW7V6u3FCEH199aj0000009,0] [APP: DIP] ClusterManager: Error
managing cluster: Failed to obtain DB connection from data source
'schedulerDS': java.sql.SQLException: Could not retrieve datasource via JNDI
url 'jdbc/schedulerDS' java.sql.SQLException: Cannot obtain connection:
driverURL = jdbc:weblogic:pool:schedulerDS, props =
{EmulateTwoPhaseCommit=false, connectionPoolID=schedulerDS,
jdbcTxDataSource=true, LoggingLastResource=false,
dataSourceName=schedulerDS}.[[
Nested Exception: java.lang.RuntimeException: Failed to setAutoCommit to true
for pool connection

AuthenticationException while connecting to OID:
[2008-11-21T00:12:08.812-08:00] [wls_ods] [ERROR] [DIP-10581] [oracle.dip]
[tid: 11] [userId: <anonymous>] [ecid: 0000Hqy6m54FW7V6u3FCEH199apO000000,0]
[APP: DIP] DIP was not able to get the context with the given details {0}[[
javax.naming.AuthenticationException: [LDAP: error code 49 - Invalid
Credentials]
```

Most exceptions are related to the scheduler or LDAP, for example:

- Could not retrieve datasource via JNDI url 'jdbc/schedulerDS' java.sql.SQLException
- javax.naming.AuthenticationException: [LDAP: error code 49 - Invalid Credentials]

Node Manager Fails to Start

If the Node Manager fails to start, ensure that you have copied the `nodemanager.domains` file from `ODIPHOST1` to `ODIPHOST2`:

```
WL_HOME/common/nodemanager/nodemanager.domains
```

Error Messages May Appear After Starting Node Manager

If you see the following error message after starting Node Manager, follow the procedure described after the error message:

```
<Dec 15, 2008 8:40:05 PM> <Warning> <Uncaught exception in server handler:
javax.net.ssl.SSLKeyException: [Security:090482]BAD_CERTIFICATE alert was
received from stbee21.example.com - 152.68.64.2155. Check the peer to
determine why it rejected the certificate chain (trusted CA configuration,
hostname verification). SSL debug tracing may be required to determine the
exact reason the certificate was rejected.> javax.net.ssl.SSLKeyException:
[Security:090482]BAD_CERTIFICATE alert was received from stbee21.example.com -
152.68.64.215. Check the peer to determine why it rejected the certificate chain
(trusted CA configuration, hostname verification). SSL debug tracing may be
required to determine the exact reason the certificate was rejected.
```

1. If you have not already done so, click **Lock & Edit** in the Change Center of the Administration Console.
2. In the left pane of the Console, expand **Servers** and **AdminServer (admin)**.
3. Select the **Configuration > SSL > Advanced Link**.
4. Select **None** for **Hostname Verification**.
5. Click **Save** to save the setting.
6. To activate these changes, in the Change Center of the Administration Console, click **Activate Changes**.
7. Restart all servers.

(Optional) Enter an example to illustrate your reference here.

1. If you have not already done so, click **Lock & Edit** in the Change Center of the Administration Console.
2. In the left pane of the Console, expand **Servers** and the name of the server that is running in ADMIN mode.
3. Select the **Control > Start/Stop** tab.
4. Select the name of the server.
5. Click **Resume**.
6. Select **Yes** to resume servers.

Configuration Changes Do Not Automatically Propagate to All Oracle Directory Integration Platform Instances in a Highly Available Topology

When you change the configuration of one Oracle Directory Integration Platform instance in a high availability topology, the configuration change does not propagate automatically to all Oracle Directory Integration Platform instances in the topology.

Use the `manageDIPServerConfig` tool to make configuration change to all Oracle Directory Integration Platform instances in the topology, ensuring the same configuration across all Oracle Directory Integration Platform instances.

See manageDIPServerConfig Utility in *Oracle Fusion Middleware Administering Oracle Directory Integration Platform*.

An Operation Cannot Be Completed for Unknown Errors Message Appears

The following error message may appear intermittently when you use the `manageSyncProfiles` command:

```
OPERATION CANNOT BE COMPLETED FOR UNKNOWN ERRORS
```

If you see this error message, start and stop the Managed Server (`wls_ods1` or `wls_ods2`). If the problem persists, repeat the copy method on the second node.

About Starting and Stopping Oracle Directory Services Components

To start and stop Oracle Directory Services Components components, see Starting and Stopping Components in *Oracle Fusion Middleware Administering Oracle Fusion Middleware*.

About Configuring Oracle Internet Directory for Maximum High Availability

This section provides high-level instructions for setting up a maximum high availability deployment for Oracle Internet Directory. This deployment includes two sites in different geographic locations. This is an active-active deployment where both sites are active at the same time when the deployment is functioning normally. If one site fails, the surviving site continues to function.

Each site includes a two-node Oracle Internet Directory cluster configuration, which provides high availability for Oracle Internet Directory. The Oracle Internet Directory cluster configuration at each site uses an Oracle Real Applications Cluster (Oracle RAC) database as the security store, which provides high availability for the database.

Oracle Internet Directory High Availability Architecture provides an introduction to the high availability Oracle Internet Directory cluster configurations.

Multimaster replication replicates data from the master site to the replica site.

Topics

- xref link to first topic
- xref link to second topic
- Overview of Maximum High Availability Oracle Internet Directory Deployment
- Overview of Replication
  This section describes the supported replication types for Oracle Internet Directory.

Overview of Maximum High Availability Oracle Internet Directory Deployment

The following figure, shows the maximum high availability deployment for Oracle Internet Directory.

The master site is located in New York and the replica site is located in Los Angeles.

Each site includes a highly available two-node Oracle Internet Directory cluster configuration that uses an Oracle RAC database as a highly available security store. Each two-node cluster has a load balancer. See Section 8.3.3, "Oracle Internet Directory High Availability Configuration Steps" for information on setting up a two-node Oracle Internet Directory cluster.

The master site in New York consists of:

- OIDHOST1 and OIDHOST2

  These are the two clustered hosts on which Oracle Internet Directory is installed.

- RAC_DB1

  The Oracle RAC database which serves as the security store for the Oracle Internet Directory instances on OIDHOST1 and OIDHOST2. Multimaster replication replicates data between RAC_DB1 in New York and RAC_DB2 in Los Angeles.

The replica site in Los Angeles consists of:

- OIDHOST3 and OIDHOST4

  These are the two clustered hosts on which Oracle Internet Directory is installed.

- RAC_DB2

  This is the Oracle RAC database which serves as the security store for the Oracle Internet Directory instances on OIDHOST3 and OIDHOST4. Multimaster replication replicates data between RAC_DB1 in New York and RAC_DB2 in Los Angeles.

Overview of Replication

This section describes the supported replication types for Oracle Internet Directory.

The following types of replication are available for Oracle Internet Directory:

- LDAP multimaster replication

  Uses the industry-standard Lightweight Directory Access Protocol Version 3 as the replication transport mechanism. Oracle recommends this protocol for replication.

- Oracle Advanced Database multimaster replication

  Uses the replication feature of Oracle Database, also called Advanced Replication.

- Two-way fan-out replication

- One-way fan-out replication

  With this replication method, the replicated data is read-only at the replica site. Fan-out uses LDAP as its transport mechanism.

For more information about the replication types for Oracle Internet Directory, see Understanding Oracle Internet Directory Replication in *Administering Oracle Internet Directory*.

For the maximum availability deployment shown in Figure 10-1, either LDAP or Oracle Advanced Database multimaster replication can be set up.

13

Configuring High Availability for Web Tier Components

Oracle Fusion Middleware's Web Tier is the architecture's outermost tier, closest to the end user.

- Oracle HTTP Server and High Availability Concepts
  Oracle HTTP Server (OHS) is the Web server component for Oracle Fusion Middleware and the key Web Tier component. It has a listener for Oracle WebLogic Server and a framework to host static pages, dynamic pages, and applications over the Web.
- Oracle HTTP Server Single-Instance Characteristics
  Oracle HTTP Server (OHS) is based on Apache infrastructure and includes Oracle modules that you can use to extend OHS core functionality.
- Oracle HTTP Server and Domains
  Oracle HTTP Server (OHS) doesn't require a WebLogic domain but you usually use it with one. Oracle recommends associating OHS with a domain so that you can incorporate OHS into the Administration Console, where you can manage and monitor it.
- Oracle HTTP Server Startup and Shutdown Lifecycle
  After Oracle HTTP Server starts, it is ready to listen for and respond to HTTP(S) requests.
- Starting and Stopping Oracle HTTP Server
  Use Fusion Middleware Control or the WebLogic Scripting Tool (WLST) to start, stop, and restart Oracle HTTP Server.
- Oracle HTTP Server High Availability Architecture and Failover Considerations
  Oracle HTTP Servers and Managed Servers reside on different hosts, behind a load balancer, in a high availability topology.
- Oracle HTTP Server Failure Protection and Expected Behaviors
  Oracle HTTP Server (OHS) has two failure types: **process failures** and **node failures**. An individual operating system process may fail. A node failure can involve failure of the entire host computer that OHS runs on.
- Configuring Oracle HTTP Server Instances on Multiple Machines
  If you use the Configuration Wizard to configure Oracle HTTP Server (OHS) and OHS is part of a domain, update the `mod_wl_ohs.conf` file for each instance.
- Configuring Oracle HTTP Server for High Availability
  To configure an example high availability deployment of Oracle HTTP Server (OHS), you must meet specific prerequisites. You can then install OHS on an additional web server, then configure and validate OHS high availability.

Oracle HTTP Server and High Availability Concepts

Oracle HTTP Server (OHS) is the Web server component for Oracle Fusion Middleware and the key Web Tier component. It has a listener for Oracle WebLogic Server and a framework to host static pages, dynamic pages, and applications over the Web.

> **Note:**
>
> For more information on working with OHS, see:
>
> - Managing Oracle HTTP Server in *Administering Oracle HTTP Server*. Includes the topics Performing Basic OHS Tasks, Creating an OHS Instance, and Managing and Monitoring Server Processes.
> - About the Oracle HTTP Server Installation in *Installing and Configuring Oracle HTTP Server*.

Oracle HTTP Server Single-Instance Characteristics

Oracle HTTP Server (OHS) is based on Apache infrastructure and includes Oracle modules that you can use to extend OHS core functionality.

OHS has these components to handle client requests

- **HTTP listener** handles incoming requests and routes them to the appropriate processing utility.
- **Modules (mods)** implement and extend OHS functionality. OHS includes many standard Apache modules. Oracle also includes modules that are specific to OHS to support OHS and OHS component integration.

OHS can also be a proxy server, both forward and reverse. A reverse proxy enables content served by different servers to appear as if it comes from one server.

Oracle HTTP Server and Domains

Oracle HTTP Server (OHS) doesn't require a WebLogic domain but you usually use it with one. Oracle recommends associating OHS with a domain so that you can incorporate OHS into the Administration Console, where you can manage and monitor it.

The `mod_wl_ohs` module handles the link to Managed Servers. You configure `mod_wl_ohs` by routing requests of a particular type, such as JSPs, or by routing requests destined to a URL to specific Managed Servers.

OHS usually front ends a cluster. In this configuration, a special `mod_wl_ohs` directive, `WebLogicCluster`, specifies a comma-separated list of cluster members.

These steps describe the `mod_wl_ohs` directive process:

1. `mod_wl_ohs` receives a request for a Managed Server then sends the request to one cluster member in the directive. At least one Managed Server must be available to fulfill the request.
2. The Managed Server receives the request, processes it, and sends a complete list of cluster members back to `mod_wl_ohs`.
3. When `mod_wl_ohs` receives the updated list, it dynamically adds previously unknown servers to the known servers list. By doing this, all future requests are load balanced across the cluster member list. The benefit is that new Managed Servers are added to a cluster without updating `mod_wl_ohs` or adding OHS.

> **Note:**
>
> The `mod_wl_ohs` directive `DynamicServerList` controls whether or not unknown servers are added to the known servers list. You must set `DynamicServerList` to `ON` to enable dynamic addition of servers.

> **Note:**
>
> When you start, you don't need to include all current Managed Servers in the `mod_wl_ohs` directive. A high availability setup requires only two cluster members in the list for the first call to work. See Configuring the WebLogic Proxy Plug-In for Oracle HTTP Server in *Oracle Fusion Middleware Using Oracle WebLogic Server Proxy Plug-Ins* for more on running an OHS high availability deployment.

> **Note:**
>
> For more on Oracle WebLogic clusters, see Introduction and Roadmap in *Using Clusters for Oracle WebLogic Server*.

Oracle HTTP Server Startup and Shutdown Lifecycle

After Oracle HTTP Server starts, it is ready to listen for and respond to HTTP(S) requests.

The request processing model is different on Microsoft Windows systems compared to UNIX systems:

- For Microsoft Windows, there is *one* parent process and *one* child process. The child process creates threads that handle client requests. The number of created threads is static and you can configure them for performance.
- For UNIX, there is *one* parent process that manages *multiple* child processes. Child processes handle requests. The parent process brings up more child processes as necessary, based on configuration.

> **Note:**
>
> For more on the OHS processing model, see Oracle HTTP Server Processing Model in *Administrator's Guide for Oracle HTTP Server*.

Starting and Stopping Oracle HTTP Server

Use Fusion Middleware Control or the WebLogic Scripting Tool (WLST) to start, stop, and restart Oracle HTTP Server.

If you plan to use WLST, you should familiarize yourself with that tool; see Getting Started Using the Oracle WebLogic Scripting Tool (WLST) in the *Oracle Fusion Middleware Administrator's Guide*.

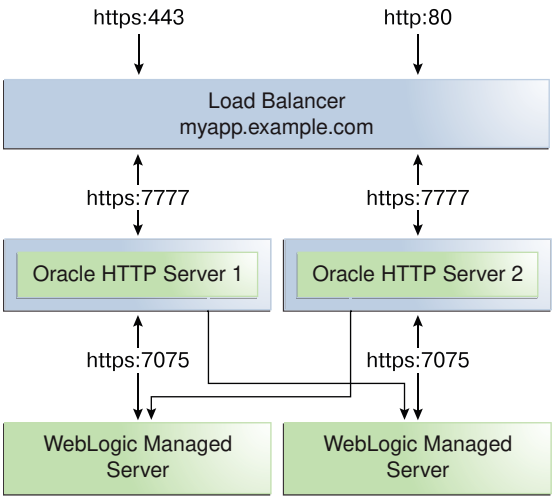
> **Note:**
>
> For steps to start and stop OHS, see Performing Basic Oracle HTTP Server Tasks in *Administrator's Guide for Oracle HTTP Server*.

Oracle HTTP Server High Availability Architecture and Failover Considerations

Oracle HTTP Servers and Managed Servers reside on different hosts, behind a load balancer, in a high availability topology.

Figure 13-1 shows two Oracle HTTP Servers behind a load balancer.

Figure 13-1 Oracle HTTP Server High Availability Architecture



The load balancer receives user requests and forwards them to connected Oracle HTTP Servers. The load balancer receives requests on standard HTTP/HTTPS ports (80/443). However, it then passes requests to Oracle HTTP Servers using completely different ports. Advantages of this setup are:

- Actual ports are hidden from users.

- Users don't have to add port numbers to the URL.

On UNIX-based systems, starting OHS with root privileges isn't mandatory. Only root can start a process that uses a port less than 1024. However, for processes that use a port number below 1024, you must use root privilege to start a process.

The load balancer routes requests to the functioning Oracle HTTP Server.

Figure 13-1 also shows how OHS distributes requests to Managed Servers. For high availability, each pair of components (OHS and Managed Servers) should reside on different host computers. Managed Servers belong to the same cluster; to load balance across a set of Managed Servers, they must belong to the same cluster.

Oracle HTTP Server Failure Protection and Expected Behaviors

Oracle HTTP Server (OHS) has two failure types: **process failures** and **node failures**. An individual operating system process may fail. A node failure can involve failure of the entire host computer that OHS runs on.

Table 13-1 OHS Failure Types and Failure Protections

Failure Type	Protection
Process	Node Manager protects and manages OHS processes. If an OHS process fails, Node Manager automatically restarts it.
Node	Load balancer in front of OHS sends a request to another OHS if the first one doesn't respond or URL pings indicate it failed.
Managed Server	If a Managed Server in a cluster fails, `mod_wl_ohs` automatically redirects requests to one of the active cluster members. If the application stores state, state replication is enabled within the cluster, which enables redirected requests access to the same state information.
Database	Typically, an issue only when using `mod_oradav` or `mod_plsql`. With Oracle RAC databases, the Oracle RAC connection determines failure characteristics. **If client connection failover is configured**, in-flight transactions roll back. Database reconnection is required. **If Transparent Application Failover (TAF) is configured**, any in-flight database write rolls back but automatic database reconnection occurs and select statements recover automatically. TAF fails over select statements only; package variables are lost. TAF, a JDBC Oracle Call Interface driver feature, enables an application to automatically reconnect to a database if the database instance the connection is made to fails. In this case, active transactions roll back.

Configuring Oracle HTTP Server Instances on Multiple Machines

If you use the Configuration Wizard to configure Oracle HTTP Server (OHS) and OHS is part of a domain, update the `mod_wl_ohs.conf` file for each instance.

The file is in the *DOMAIN_HOME*`/config/fmwconfig/components/OHS/`*componentName* directory. Restart the Administration Server to propagate changes to all OHS instances in the domain, even if they reside on a different host.

> **Note:**
>
> See Configuring mod_wl_ohs.conf for more information on the `mod_wl_ohs` file.

> **Note:**
>
> If you install and configure OHS instances in separate domains, you must manually copy changes to other Oracle HTTP Servers. You must verify that the changes apply to all OHS instances and that they are synchronized.

Configuring Oracle HTTP Server for High Availability

To configure an example high availability deployment of Oracle HTTP Server (OHS), you must meet specific prerequisites. You can then install OHS on an additional web server, then configure and validate OHS high availability.

- Prerequisites to Configure a Highly Available OHS
  You must meet certain prerequisites before configuring a high availability Oracle HTTP Server deployment.
- Installing and Validating Oracle HTTP Server on WEBHOST2
  You install OHS then validate the install.
- Configuring and Validating an OHS High Availability Deployment
  To configure and validate the OHS high availability deployment, update `mod_wl_ohs.conf` and then use test URLs to validate OHS configuration.

Prerequisites to Configure a Highly Available OHS

You must meet certain prerequisites before configuring a high availability Oracle HTTP Server deployment.

- Load Balancer Prerequisites
  To distribute requests against Oracle HTTP Server, you must use an external load balancer to distribute HTTP(S) requests between available Oracle HTTP Servers.
- Configuring Load Balancer Virtual Server Names and Ports
  In an OHS installation, virtual servers are configured for HTTP connections, which are distributed across the HTTP servers.
- Managing Load Balancer Port Numbers
  Many Oracle Fusion Middleware components and services use ports. As an administrator, you must know the port numbers that services use and ensure that two services don't use the same port number on your host computer.
- Installing and Validating Oracle HTTP Server on WEBHOST1
  After you install OHS, validate its installation.
- Creating Virtual Host(s) on WEBHOST1
  For each virtual host or site name that you use, add an entry to the OHS configuration.

- Configuring mod_wl_ohs.conf
  After you install and configure OHS, link it to any defined Managed Servers by editing the `mod_wl_ohs.conf` file.
- Configuring mod_wl_conf if you use SSL Termination
  If you use SSL termination AND route requests to WebLogic, you must take additional configuration steps.

Load Balancer Prerequisites

To distribute requests against Oracle HTTP Server, you must use an external load balancer to distribute HTTP(S) requests between available Oracle HTTP Servers.

If you have an external load balancer, it must have features that Third-Party Load Balancer Requirements describes.

Configuring Load Balancer Virtual Server Names and Ports

In an OHS installation, virtual servers are configured for HTTP connections, which are distributed across the HTTP servers.

If your site serves requests for HTTP and HTTPS connections, Oracle recommends that HTTPS requests terminate at the load balancer and pass through as HTTP requests. To do this, the load balancer should be able to perform the protocol conversion and must be configured for persistent HTTP sessions.

This example configuration assumes that the load balancer is configured as:

- **Virtual Host:** `Myapp.example.com`
- **Virtual Port:** `7777`
- **Server Pool:** `Map`
- **Server:** `WEBHOST1`, `Port 7777`, `WEBHOST2`, `Port 7777`

Managing Load Balancer Port Numbers

Many Oracle Fusion Middleware components and services use ports. As an administrator, you must know the port numbers that services use and ensure that two services don't use the same port number on your host computer.

Most port numbers are assigned during installation. It is important that any traffic going from Oracle HTTP Servers to Oracle WebLogic Servers has access through any firewalls.

Installing and Validating Oracle HTTP Server on WEBHOST1

After you install OHS, validate its installation.

To install OHS on WEBHOST1, see the steps in Installing the Oracle HTTP Server Software in *Oracle Fusion Middleware Installing and Configuring Oracle HTTP Server*.

Validate the installation using the following URL to access the OHS home page:

```
http://webhost1:7777/
```

Creating Virtual Host(s) on WEBHOST1

For each virtual host or site name that you use, add an entry to the OHS configuration.

Create a file named `virtual_hosts.conf` in the `DOMAIN_HOME/config/fmwconfig/components/OHS/ohs_component_name/moduleconf` directory as follows:

```
NameVirtualHost *:7777
<VirtualHost *:7777>
 ServerName http://myapp.example.com:80
 RewriteEngine On
 RewriteOptions inherit
 UseCanonicalName On
</VirtualHost>
```

If you are using SSL/SSL Termination (*):

```
NameVirtualHost *:7777
<VirtualHost *:7777>
 ServerName https://myapp.example.com:443
 RewriteEngine On
 RewriteOptions inherit
 UseCanonicalName On
</VirtualHost>
```

> **Note:**
>
> You can also use Fusion Middleware Control to create virtual hosts. See Wiring Components Together in *Administering Oracle Fusion Middleware*.

Configuring mod_wl_ohs.conf

After you install and configure OHS, link it to any defined Managed Servers by editing the `mod_wl_ohs.conf` file.

The file is in *DOMAIN_HOME*`/config/fmwconfig/components/OHS/`*componentName* directory.

See Configuring the WebLogic Proxy Plug-In for Oracle HTTP Server in *Oracle Fusion Middleware Using Oracle WebLogic Server Proxy Plug-Ins 12.1.2* for more about editing the `mod_wl_ohs.conf` file.

> **Note:**
>
> You can also use Fusion Middleware Control to link OHS to Managed Servers. See Wiring Components Together in *Administering Oracle Fusion Middleware*

The following example shows `mod_wl_ohs.conf` entries:

```
LoadModule weblogic_module PRODUCT_HOME/modules/mod_wl_ohs.so
```

```
<IfModule mod_weblogic.c>
 WebLogicCluster apphost1.example.com:7050, apphost2.example.com:7050
 MatchExpression *.jsp
 </IfModule>

<Location /weblogic>
 SetHandler weblogic-handler
 WebLogicCluster apphost1.example.com:7050,apphost2.example.com:7050
 DefaultFileName index.jsp
</Location>

<Location /console>
 SetHandler weblogic-handler
 WebLogicCluster apphost1.example.com
 WebLogicPort 7003
</Location>
```

These examples show two different ways to route requests to Managed Servers:

- The `<ifModule>` block sends any requests ending in *.jsp to the WebLogic Managed Server cluster located on APPHOST1 and APPHOST2.
- The `<Location>` block sends any requests with URLs that have a `/weblogic` prefix to the Managed Server cluster located on APPHOST1 and APPHOST2.

Configuring mod_wl_conf if you use SSL Termination

If you use SSL termination AND route requests to WebLogic, you must take additional configuration steps.

To configure mod_wl_conf if you use SSL termination:

1. In the WebLogic console, verify that WebLogic Plugin Enabled is set to true, either at the domain, cluster, or Managed Server level.
2. Add these lines to the Location block, which directs requests to Managed Servers:

   ```
   WLProxySSL ON
   WLProxySSLPassThrough ON
   ```

For example:

```
<Location /weblogic>
 SetHandler weblogic-handler
 WebLogicCluster apphost1.example.com:7050,apphost2.example.com:7050
 WLProxySSL On
 WLProxySSLPassThrough ON
 DefaultFileName index.jsp
</Location>
```

After you enable the WebLogic plugin, restart the Administration Server.

Installing and Validating Oracle HTTP Server on WEBHOST2

You install OHS then validate the install.

To install Oracle HTTP Server on WEBHOST2, see Installing the Oracle HTTP Server Software in *Oracle Fusion Middleware Installing and Configuring Oracle HTTP Server*.

Validate the installation on WEBHOST2 by using the following URL to access the Oracle HTTP Server home page:

`http://webhost2:7777/`

Configuring and Validating an OHS High Availability Deployment

To configure and validate the OHS high availability deployment, update `mod_wl_ohs.conf` and then use test URLs to validate OHS configuration.

- Configuring Virtual Host(s) on WEBHOST2
  Update the `mod_wl_ohs.conf` file located in *DOMAIN_HOME*`/config/fmwconfig/components/OHS/`*componentName* directory.
- Validating the Oracle HTTP Server Configuration
  You validate the OHS configuration by using specific URLs.

Configuring Virtual Host(s) on WEBHOST2

Update the `mod_wl_ohs.conf` file located in *DOMAIN_HOME*`/config/fmwconfig/components/OHS/`*componentName* directory.

You must then restart the Administration Server to propagate changes to all OHS instances in the domain.

Validating the Oracle HTTP Server Configuration

You validate the OHS configuration by using specific URLs.

`http://myapp.example.com/`

`https://myapp.example.com` (if using SSL/SSL termination)

`http://myapp.example.com:7777/weblogic`

14

Configuring High Availability for SOA Components

Keep the following in mind when you deploy SOA in a high availability environment.

- About Working with Human Tasks in SOA Composer
  In a high availability SOA environment, the HTTP session expires if you select a human task in the SOA Composer navigation panel.
- About Composer Failover
  In a high availability environment, you lose session state in Composer if failover occurs. Composer is accessible on the secondary server but you must log in again and create a fresh session.

About Working with Human Tasks in SOA Composer

In a high availability SOA environment, the HTTP session expires if you select a human task in the SOA Composer navigation panel.

About Composer Failover

In a high availability environment, you lose session state in Composer if failover occurs. Composer is accessible on the secondary server but you must log in again and create a fresh session.

15

Configuring High Availability for Oracle WebCenter Components

There are special considerations to keep in mind when you configure high availability for Oracle WebCenter.
For more information on WebCenter, see the installation guide for the WebCenter product you are working with. For WebCenter Portal, you can also refer toUnderstanding the WebCenter Portal Enterprise Deployment Topology in *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle WebCenter Portal*.

- About Extending WebCenter Content: Inbound Refinery Components
  WebCenter Content: Inbound Refinery components don't use WebLogic Server clustering for scale up and failover. You can deploy and configure multiple, independent Inbound Refinery Managed Servers for WebCenter Content to use.
- About WebCenter Content Scaleup and Ports
  When you scale up WebCenter Content, the scaled-up server uses a port which is different from the port that you configured in the Configuration Wizard.
- About Creating WebCenter Portal Components on Multiple Nodes
  In a WebCenter Portal high availability setup, you might see a delay of data appearing on the second node due to a 10-minute default refresh interval. For example, you may see this if you are trying to access a new Portal component that you just created on Node 2 if the Node 1 is processing the request.
- About Creating a WebCenter Capture Domain with Oracle SOA
  If you create a WebCenter Capture domain with an Oracle SOA component, you must select the JRF-MAN-SVR and WSMPM-MAN-SVR server groups. These server groups ensure that the oracle JRF and Oracle Web Services Manager (OWSM) services target the Managed Servers you are creating.
- About Scaling Out WebCenter Capture and Configuring OWSM
  OWSM uses cross-component wiring to auto-discover the Policy Manager in a domain. When you use the Configuration Wizard to create or update a domain that includes OWSM, Policy Manager URLs publish to the Local Service table.
- About WebCenter Sites Component Connections
  WebCenter Sites requires an external component that is up and running when trying to establish connections. If an external component isn't up and running or the connection has timed out, WebCenter Sites doesn't reestablish the connection.
- About WebCenter Sites and Multicast
  WebCenter Sites relies on multicast support to provide inCache replication across nodes in a clustered environment.

About Extending WebCenter Content: Inbound Refinery Components

WebCenter Content: Inbound Refinery components don't use WebLogic Server clustering for scale up and failover. You can deploy and configure multiple, independent Inbound Refinery Managed Servers for WebCenter Content to use.

About WebCenter Content Scaleup and Ports

When you scale up WebCenter Content, the scaled-up server uses a port which is different from the port that you configured in the Configuration Wizard.

In the WebCenter Content configuration file, you must specify the IntradocServerPort value for each Managed Server instance. The port for each Managed Server instance that resides on the one machine must be unique. For example, if `Content_server1` and `Content_server2` are on the same machine, the configuration file should look similar to the following:

```
IntradocServerPort=4444
IntradocServerPort.UCM_serverd1=4444
IntradocServerPort.UCM_serverd2=4445
```

About Creating WebCenter Portal Components on Multiple Nodes

In a WebCenter Portal high availability setup, you might see a delay of data appearing on the second node due to a 10-minute default refresh interval. For example, you may see this if you are trying to access a new Portal component that you just created on Node 2 if the Node 1 is processing the request.

About Creating a WebCenter Capture Domain with Oracle SOA

If you create a WebCenter Capture domain with an Oracle SOA component, you must select the JRF-MAN-SVR and WSMPM-MAN-SVR server groups. These server groups ensure that the oracle JRF and Oracle Web Services Manager (OWSM) services target the Managed Servers you are creating.

About Scaling Out WebCenter Capture and Configuring OWSM

OWSM uses cross-component wiring to auto-discover the Policy Manager in a domain. When you use the Configuration Wizard to create or update a domain that includes OWSM, Policy Manager URLs publish to the Local Service table.

OWSM Agent is automatically wired to the OWSM Policy Manager using endpoint entries published to the Local Service table. If, however, you change the domain using tools other than the Configuration Wizard (such as WebLogic Administration Console, Fusion Middleware Control, or WLST), any changes to the Policy Manager URL automatically publish to the Local Service table but the OWSM Agent client isn't automatically bound to the new URL. In this case, you must manually bind OWSM Agent to the Policy Manager URL.

For more, see Verifying Agent Bindings Using Fusion Middleware Control in *Oracle Fusion Middleware Securing Web Services and Managing Policies with Oracle Web Services Manager*.

About WebCenter Sites Component Connections

WebCenter Sites requires an external component that is up and running when trying to establish connections. If an external component isn't up and running or the connection has timed out, WebCenter Sites doesn't reestablish the connection.

WebCenter Sites verifies that database connections are active and reestablishes connections only when they are active.

About WebCenter Sites and Multicast

WebCenter Sites relies on multicast support to provide inCache replication across nodes in a clustered environment.

For more about the inCache support in WebCenter Sites, see Using the inCache Framework in *Oracle Fusion Middleware Administering Oracle WebCenter Sites*.

16

Configuring High Availability for Other Components

This section describes information unique to certain component products.
For this release, this section includes the following topics:

- Deploying Oracle Data Integrator
  Review information in this section, which describes considerations for configuring Oracle Data Integrator repository connections to Oracle Real Application Clusters:
- Deploying Oracle Application Development Framework
  Review this section for special considerations to deploy Oracle Application Development Framework (ADF).
- Deploying BI
  Keep the following in mind as you deploy Oracle Business Intelligence (BI).
- Deploying Forms
  Keep the following in mind as you deploy Forms.
- Deploying Reports
  Reports has the certain considerations you need to know about for a high availability set up.
- Deploying Oracle Business Process Management
  Keep the following in mind when you deploy Oracle Business Process Management in a high availability environment.

Deploying Oracle Data Integrator

Review information in this section, which describes considerations for configuring Oracle Data Integrator repository connections to Oracle Real Application Clusters:

- Oracle RAC Retry Connectivity for Source and Target Connections
  When you configure Oracle Data Integrator (ODI) Oracle Real Application Clusters (RAC) connections, Oracle RAC retry is supported for the ODI master or ODI work repository.
- Configuring ODI Repository Connections to Oracle RAC
  When you create an ODI repository using Repository Creation Utility (RCU), you specify the work repository connection JDBC URL. RCU stores the URL in the master repository contents. If a work repository JDBC URL is a single node URL, Oracle recommends that you modify the URL to include the Oracle Real Application Clusters (Oracle RAC) failover address.
- About Oracle Data Integrator Scheduler Node Failure
  If a WebLogic Server failover occurs, the other WebLogic Server instance becomes the scheduler. A Coherence cache handles the scheduler lifecycle. Locking guarantees the scheduler uniqueness, and event notification provides scheduler migration.

Oracle RAC Retry Connectivity for Source and Target Connections

When you configure Oracle Data Integrator (ODI) Oracle Real Application Clusters (RAC) connections, Oracle RAC retry is supported for the ODI master or ODI work repository.

ODI uses transactional connections to source and target connections while running ODI scenarios. For these source and target connections, ODI doesn't support RAC retry connectivity. You can't migrate these transactions to another node in Oracle RAC.

Configuring ODI Repository Connections to Oracle RAC

When you create an ODI repository using Repository Creation Utility (RCU), you specify the work repository connection JDBC URL. RCU stores the URL in the master repository contents. If a work repository JDBC URL is a single node URL, Oracle recommends that you modify the URL to include the Oracle Real Application Clusters (Oracle RAC) failover address.

- If Oracle RAC is *not* configured with Single Client Access Name (SCAN), you can provide details of the Oracle RAC instances. In the work repository JDBC URL field, enter the Oracle RAC connectivity address in the format *host*:*port*. See the following example.
- If Oracle RAC is configured with SCAN, provide Oracle RAC instance details with the SCAN address.

The following example shows the JDBC URL format to connect to an Oracle RAC with two hosts when it doesn't use SCAN:

```
jdbc:oracle:thin:(DESCRIPTION =(LOAD_BALANCE=ON) (ADDRESS =(PROTOCOL =tcp)
(HOST =host1)(PORT =port1)) (ADDRESS =(PROTOCOL =tcp)(HOST =host2)
(PORT =port2)) (CONNECT_DATA =(SERVER=dedicated)
(SERVICE_NAME=service_name)))
```

See Creating a Work Repository in *Administering Oracle Data Integrator*.

About Oracle Data Integrator Scheduler Node Failure

If a WebLogic Server failover occurs, the other WebLogic Server instance becomes the scheduler. A Coherence cache handles the scheduler lifecycle. Locking guarantees the scheduler uniqueness, and event notification provides scheduler migration.

When an agent restarts and computes its schedule, it takes into account schedules in progress, which automatically continue their execution cycle beyond the server startup time. New sessions trigger as if the scheduler never stopped. Stale sessions move to an error state and remain in that state when they restart.

In an Oracle Data Integrator Agent cluster, if the Agent node that is the scheduler node fails, another node in the cluster takes over as the scheduler node. The new scheduler node reinitializes and runs all schedules from that point forward.

If a scheduled scenario with a repeatable execution cycle is running when the node crashes, the scenario doesn't continue its iterations on the new scheduler node from the point at which the scheduler node failed. For example, if a scheduled scenario is configured to repeat the execution 10 times after an interval of two minutes and the

scheduler node fails during the third execution, the new scheduler node doesn't continue running the scenario for the next eight executions.

Deploying Oracle Application Development Framework

Review this section for special considerations to deploy Oracle Application Development Framework (ADF).

> **Note:**
>
> For more on ADF, see:
>
> - Oracle ADF Key Concepts in *Understanding the Oracle Application Development Framework*
> - *Oracle Fusion Middleware Administering Oracle ADF Applications*

- Oracle JRF Asynchronous Web Services (Pinned Service Behavior)

Oracle JRF Asynchronous Web Services (Pinned Service Behavior)

When you use Oracle JRF Asynchronous Web Services, the asynchronous web service is pinned to a service and doesn't fail over. When you use a reliability protocol such as WS-RM, the higher-level protocol reconnects to a new server after a failure.

For more on Oracle JRF Asynchronous Web Services, see Oracle JRF and ADF TemplatesOracle JRF Template in *Domain Template Reference*.

Deploying BI

Keep the following in mind as you deploy Oracle Business Intelligence (BI).

- About BI Session Failover
  If a BI Managed Server and/or host crashes, a user may need to log in again. This depends on which application they are using at the time of the crash and whether or not SSO is in use.
- About BI Essbase
  Essbase doesn't support a high availability configuration. If a server fails, there is no loss of state; you can recover from a failure by redploying Essbase Cube.
- About BI Studio
  Studio doesn't support a high availability configuration. Oracle recommends performing xml import/export on a regular basis. This is the best practice for Studio recovery from a catalog failure.
- About Specifying Ports for Multiple Node Managers
  If you have more than one node manager per machine, verify that you specify your ports.
- About RAC Database Post Installation Configuration
  Oracle BI requires additional configuration steps for whole server migration after installation.
- About Scaling Out BI Publisher
  Completing BI scale out requires tasks in addition to typical scale out steps.

About BI Session Failover

If a BI Managed Server and/or host crashes, a user may need to log in again. This depends on which application they are using at the time of the crash and whether or not SSO is in use.

About BI Essbase

Essbase doesn't support a high availability configuration. If a server fails, there is no loss of state; you can recover from a failure by redploying Essbase Cube.

About BI Studio

Studio doesn't support a high availability configuration. Oracle recommends performing xml import/export on a regular basis. This is the best practice for Studio recovery from a catalog failure.

About Specifying Ports for Multiple Node Managers

If you have more than one node manager per machine, verify that you specify your ports.

See About Node Manager Configuration in a Typical Enterprise Deployment in *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle Business Intelligence*.

About RAC Database Post Installation Configuration

Oracle BI requires additional configuration steps for whole server migration after installation.

See Using Whole Server Migration and Service Migration in an Enterprise Deployment in Oracle Fusion Middleware Enterprise Deployment Guide for Oracle Business Intelligence for these steps.

About Scaling Out BI Publisher

Completing BI scale out requires tasks in addition to typical scale out steps.

Oracle BI requires additional steps after you follow scale out steps in Scaling Out a Topology (Machine Scale Out). Oracle BI requires you to change `setDomainEnv.sh` to the updated singleton-data-directory setting (SDD).

To complete Oracle BI scale out:

1. Move the SDD from local to shared storage so that all hosts can access it using the same path. For example, move it to:

   `DOMAIN_HOME/bidata/net/yoursystem/scratch/sdd`

2. Open DOMAIN_HOME/config/fmwconfig/bienv/core/bi-environment.xml (element bi:singleton-data-directory).
3. Change the `xdo.server.config.dir` path to refer to the new SDD path you just created.

4. Restart the server.

Deploying Forms

Keep the following in mind as you deploy Forms.

- About Recovering from Forms HTTP Session Failover

About Recovering from Forms HTTP Session Failover

If a Forms HTTP session fails, you must reconnect and restart your session with the Forms application.

Deploying Reports

Reports has the certain considerations you need to know about for a high availability set up.

- About Scaling Up in Reports
  If you scale up Reports components, Oracle recommends that you bring down all nodes and then restart them when configuration is complete.
- About Reports Multicast Communication
  Reports cluster members or individual clients use multicast to discover other nodes. There is no workaround to using multicast.
- About Reports Shared-File Based Cache
  Reports has shared file based cache as a singleton. If the cache fails, high availability also fails.
- About Reports Database Service Failure
  Reports components can tolerate database failure. Reports retries the database connection three times. After the database is up, you must run Reports again.
- About Reports OID/Shared Cache File System Failure
  Reports doesn't have retries for OID/Shared cache file system failures. There is no workaround. After the external system is up, you must run Reports again.

About Scaling Up in Reports

If you scale up Reports components, Oracle recommends that you bring down all nodes and then restart them when configuration is complete.

See Starting and Stopping Oracle Reports Server in *Oracle Fusion Middleware Publishing Reports to the Web with Oracle Reports Services*.

About Reports Multicast Communication

Reports cluster members or individual clients use multicast to discover other nodes. There is no workaround to using multicast.

About Reports Shared-File Based Cache

Reports has shared file based cache as a singleton. If the cache fails, high availability also fails.

There is no workaround. If shared-file based cache fails, you must restart Reports servers.

About Reports Database Service Failure

Reports components can tolerate database failure. Reports retries the database connection three times. After the database is up, you must run Reports again.

About Reports OID/Shared Cache File System Failure

Reports doesn't have retries for OID/Shared cache file system failures. There is no workaround. After the external system is up, you must run Reports again.

Deploying Oracle Business Process Management

Keep the following in mind when you deploy Oracle Business Process Management in a high availability environment.

About BP Composer and High Availability

In a high availability environment, you lose session state in BPM Composer if failover occurs. This causes loss of work; any in-progress edits are lost. BPM Composer is accessible on the secondary server but you must log in again and create a fresh session.